TM-738/030/00

The Equivalence of Context Limited Grammars to Context Free Grammars

SCIENTIFIC REPORT NO. 4

Thomas N Hibbard

16 January 1967

Contract Monitor Thomas V Griffiths
Data Sciences Labortory

Prepared for
AIR FORCE CAMBRIDGE RESEARCH LABORATORIES
OFFICE OF AEROSPACE RESEARCH, USAF
BEDFORD, MASSACHUSETTS
and
AIR FORCE OFFICE OF SCIENTIFIC RESEARCH
OFFICE OF AEROSPACE RESEARCH, USAF
ARLINGTON, VIRGINIA

Monitored by R Swanson, SRIR (AFOSR) Supported by SDC's Independent Research Program, Contract F1962867C0008, Programming (Algorithmic) Languages, Project No 5632, Task No 563205, and Grant No AF AFOSR 1203 67

Work Unit No 56320501

66

AFCRL-67-0434

TM-738/030/00

SDC
TM SERIES

The work reported herein was supported by SDC; contract F1962867C0008, CRL - Algorithmic Languages Program, project no. 5632, task no. 563205; and grant AF-AFOSR-1203-67. Work Unit No. 56320501.

SCIENTIFIC REPORT NO. 4

The Equivalence of Context Limited Grammars to Context Free Grammars

by

Thomas N. Hibbard

16 January 1967

Monitored by: T. Griffiths (AFCRL)
Data Sciences Laboratory
R. Swanson, SRIR (AFOSR)

Prepared for

AIR FORCE CAMBRIDGE RESEARCH LABORATORIES
OFFICE OF AEROSPACE RESEARCH, USAF
BEDFORD, MASSACHUSETTS

and

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH
OFFICE OF AEROSPACE RESEARCH, USAF
ARLINGTON, VIRGINIA

SYSTEM DEVELOPMENT CORPORATION
2500 COLORADO AVE.
SANTA MONICA
CALIFORNIA
90406

ABSTRACT

A grammar G is context limited if there exists a partial ordering on the alphabet of G under which, for every production $\alpha \rightarrow \beta$ of G, every letter of α is smaller than some letter of β. It is proved that the languages generated by context limited grammars are just the context free languages. Unambiguity of general grammars is defined and discussed carefully, preparatory to proving that the languages generated by unambiguous context limited grammars are just the unambiguous context free languages.

THE EQUIVALENCE OF CONTEXT LIMITED GRAMMARS TO CONTEXT FREE GRAMMARS[1]

A phrase structure grammar [3, p.8] is here called <u>context</u> <u>limited</u> if there exists a partial ordering on the alphabet of the grammar, under which, for every production $\alpha \rightarrow \beta$ of the grammar, every letter of α is strictly smaller than some letter of β.

We will prove

(1) that the context limited grammars generate all and only context free languages; and

(2) that the unambiguous context limited grammars generate all and only unambiguous context free languages.

The bulk of the effort goes to (2) because, although the intuitive notion of unambiguity is clear enough, it has not previously been defined formally for phrase structure grammars in general.[2] The notion of the ambiguity of a grammar is just that there be two essentially different ways of generating the same word. Section 2 is devoted to formalizing this term "essentially

(1) Research sponsored in part by the Air Force Cambridge Research Laboratories, Office of Aerospace Research, USAF, under Contract F1962867C00008, and by the Air Force Office of Scientific Research, Office of Aerospace Research, USAF, under AFOSR Grant No. AF-AFOSR-1203-67.

(2) This is because there has been no need for such a definition. With respect to the most general class of phrase structure grammars, there are no inherently ambiguous grammars, and it is not yet known whether the corresponding assertion for context sensitive grammars holds or not.

different," to establishing a few basic facts about the formulation, including the fact that it reduces to the usual one in the context free case, and to proving some technical lemmas for use in Section 3.

In Section 3 the context limited grammars are defined formally and their two main properties, indicated above, are proved (Theorems 1 and 2).

Section 1 explains the notation.

Most of the material presented here appeared first in the author's doctoral dissertation [10].

1. Notation

The notation is drawn mainly from [3], with the exceptions now to be noted.

Grammar will mean phrase structure grammar [3, p.8].

Let $G = (V,\Sigma,P,\sigma)$ be a grammar. We will refer to the letters in Σ as external letters (to avoid conflict with another natural use of the term "terminal") and to those in $V-\Sigma$ as internal letters.

$\alpha \vdash_{G} \beta$ will be used instead of $\alpha \overset{*}{\underset{G}{\Rightarrow}} \beta$.

A complete G-derivation is a G-derivation beginning with σ and ending with a word in Σ^*.

σ is the initial or start letter of G.

2. G-graphs and Ambiguity

The first objective is to develop the notation for talking about ambiguity, to establish a few simple facts about it, and to set forth the definition of ambiguity.

The definition of ambiguity to be given is equivalent to the one implicit in Griffiths' paper [9]. However, using Griffiths' definition would present as much of a problem in establishing fundamental properties as does the present one. Moreover, we feel that the present construction embodies in a more direct manner the intuitive notion of ambiguity; for, whereas Griffiths chooses a canonical derivation from a set of equivalent derivations, we are going to deal with a structure possessed in common by all the derivations in an equivalence class.

The basic entities are the <u>production graph</u> and the <u>labeled production graph</u>. The notion of a production graph is represented accurately by strings of beads hung together as in Figure 1. The numbers beside the beads are only for reference and do not represent part of the graph. The string (1,2,3,4,5,6) is the <u>initial string</u> of the graph. The string (1,7,8,10,15,16,14) is the <u>terminal</u> string, for these are the exposed beads in their proper left-to-right order, all other beads having been "covered" by hanging a new string across them. If we wanted to specify this graph without drawing a figure, we could use a notation such as the following:

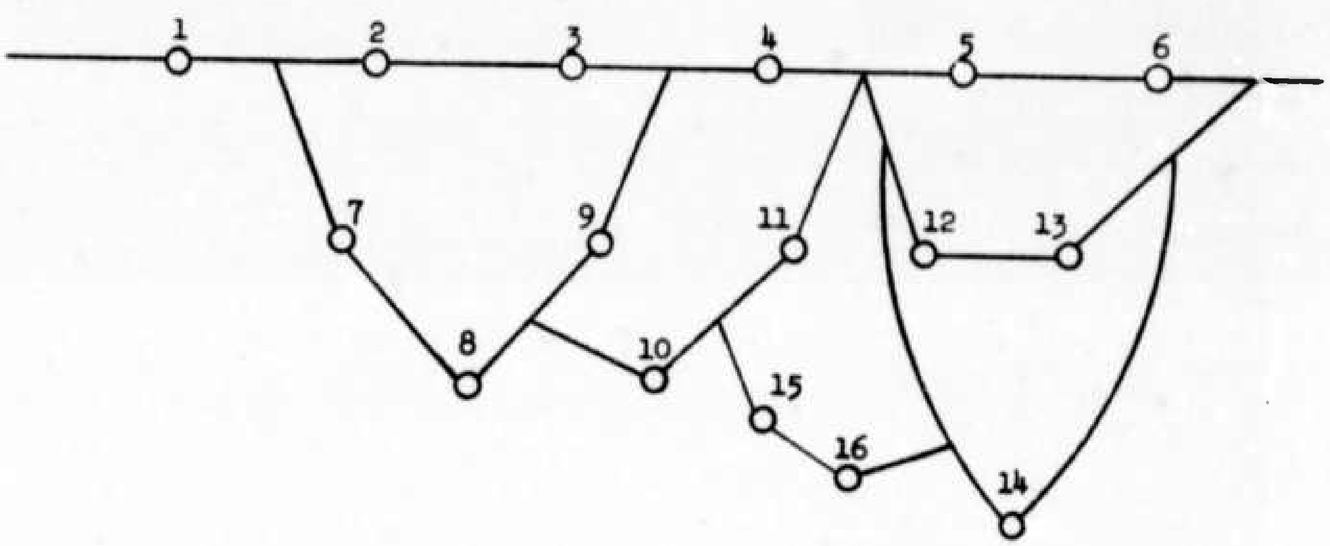

Figure 1. Schematic Representation of a Production Graph

Initial string: 1,2,3,4,5,6

Cover 2,3 by 7,8,9
5,6 by 12,13
9,4 by 10,11
12,13 by 14
11 by 15,16

Thus a production graph is specified by specifying an initial string and a set of ordered pairs of strings. Note that not every set of ordered pairs of strings makes sense: every first string must at some stage of the construction be a segment of the terminal string at that stage, and at the same stage the accompanying second string must consist of newly introduced beads. These conditions would be taken care of automatically in the process of actually hanging the beads, but must be made explicit in the formal definition.

The beads will be called "nodes"; the ordered pairs of strings will be called "productions."

By associating a letter of some alphabet with each node of a production graph we construct a labeled production graph. If G is a grammar and T is a labeled production graph, if each production of T forms a production of G when the labels replace the node names, and if every label is in the alphabet of G, then T is a "G-graph." Every G-graph represents at least one (possibly more) G-derivation. If G-graphs representing complete derivations are unique (up to congruence) for each derived word, then G will be called unambiguous.

The remainder of this section is devoted to giving precise formulations for the above notions and making sure that these formulations have the

properties we want. We will first confine our attention to production graphs, then consider labeled graphs and G-graphs.

2.1. Production graphs.

By a string we will mean a (possibly null) finite nonrepeating sequence of elements of some set. If τ and σ are strings, then $\tau\sigma$ will denote the concatenation of the two. Of course $\tau\sigma$ need not be a string just because τ and σ are, but we adopt the convention that when we write $\tau\sigma$ we mean it to be a string.

σ is a substring of τ if $\tau = \tau_1\sigma\tau_2$.

A (graphical) production is an ordered pair (σ, τ) where σ and τ are strings with no element in common and neither σ nor τ is null.

If Φ is a set of productions, then, by the nodes of Φ, we mean the elements in the strings in the productions in Φ.

The term "production graph" will denote a structure associated with each of two derivations when the two derivations are "essentially the same." A production graph will consist simply of a set of productions together with an "initial string," which together have certain properties. These properties are most easily expressed by an inductive definition.

In the definition of production graph, given next, it will be convenient to include the definition of the terminal string of the production graph, so that we will actually be defining an ordered triple, "a production graph (Φ,μ) with terminal string τ." It will be shown shortly that Φ and μ entail a unique τ so that production graphs can be given as ordered pairs.

Let $\emptyset$ denote the empty set.

DEFINITION.

PG1. If μ is a string, then $(\emptyset,\mu)$ is a production graph with terminal string μ.

PG2. If (Φ,μ) is a production graph with terminal string $\tau_1\tau_2\tau_3$, and if ρ is a string containing no node of Φ and no node of μ, then $(\Phi \cup \{\tau_2, \rho)\},\mu)$ is a production graph with terminal string $\tau_1 \; \rho \; \tau_3$.

PG3. (Φ,μ) is a production graph with terminal string τ only if required by PG1 and PG2.

If (Φ,μ) is a production graph, then we call μ the initial string of (Φ,μ).

If (Φ,μ) and ρ are as in PG2, and if T denotes (Φ,μ), then $T + (\tau_2, \rho)$ denotes the production graph $(\Phi \cup \{(\tau_2, \rho)\},\mu)$.

We are going to rely heavily on certain properties of production graphs. All of these properties are just what we would expect them to be, but just elusive enough to require proof.

Our first task is to show that every production graph has a unique terminal string. We anticipate this proof in the following definition:

A terminal node of a production graph is a node belonging to the terminal string of the production graph.

For the time being, read "a terminal string" for "the terminal string" in the previous definition.

PROPOSITION 2.1.1. Let T be a production graph. A node p of T is a terminal node of T if and only if p does not occur in the first string of any production of T. Every terminal node of T occurs in every terminal string of T.

Proof. The assertion is obvious for graphs with no productions. Suppose the assertion holds for graphs with n productions, and let T be a graph with n + 1 productions. Let p occur in a terminal string of T. Then there exists a graph T_o with n productions and terminal string $\rho_1\sigma\rho_2$, and a production (σ,τ) such that $T = T_o + (\sigma,\tau)$, and such that p is in $\rho_1\tau\rho_2$. If p is in τ then p is not in σ and p is not a node of T_o (for all the nodes of τ must be foreign to T_o in order to apply PG2). Hence p does not occur in the first string of any production of T. If p is in ρ_1 or ρ_2, then p is a terminal node of T_o, hence by inductive hypothesis occurs in the first string of no production of T_o, and p is not in σ because $\rho_1\sigma\rho_2$ is a string. Thus p occurs in the first string of no production of T.

Conversely, suppose p occurs in the first string of no production of T. A graph with no productions can only come via PG1, where it is clear that every node is in the one terminal string. Suppose it is true in graphs with n productions that every node such as p belongs to every terminal string. Let T be a graph with n + 1 productions. Any terminal string of T is given by a graph T_o with terminal string $\rho_1\sigma\rho_2$ and a production (σ,τ), with $T = T_o + (\sigma,\tau)$, where T_o has n productions. If p is in τ then p occurs in the terminal string. If p is not in τ then p is a node of T_o. Every production of T_o is a production of T, so p occurs in the first string of no production of T_o. Hence p occurs in every terminal string of T_o, in particular in

$\rho_1\sigma\rho_2$. p is not in σ for then p would occur in the first string of (σ,τ). Hence p is in ρ_1 or ρ_2, hence in $\rho_1\tau\rho_2$.

PROPOSITION 2.1.2. No node of a production graph T occurs in the second string of more than one production of T.

Proof. The assertion holds if there are no productions. Assume the assertion holds for all graphs with n productions. Let $T = T_o + (\sigma,\tau)$ where T_o has n productions. The nodes of τ are foreign to T_o so no node of τ can occur in the second string of two distinct productions of T. But for any other node to do so would require it to do so in T_o, contrary to the induction hypothesis.

PROPOSITION 2.1.3. Let T be a production graph and (σ,τ) a production of T such that all the nodes of τ are terminal nodes of T. Then τ is a substring of every terminal string of T, and for every terminal string $\rho_1\tau\rho_2$ of T there exists a unique production graph T' with terminal string $\rho_1\sigma\rho_2$ such that $T = T' + (\sigma,\tau)$.

Notation. The unique T' such that $T = T' + (\sigma,\tau)$ will be denoted $T - (\sigma,\tau)$.

Proof. The assertion is vacuously true for graphs with no productions. Assume it holds for graphs with n productions. Let T have n + 1 productions and satisfy the hypothesis with respect to σ,τ. Let λ be a terminal string of T. There exists a graph T_o with terminal string $\lambda_1\sigma_o\lambda_2$ such that $T = T_o + (\sigma_o,\tau_o)$ and such that $\lambda_1\tau_o\lambda_2 = \lambda$. If $(\sigma_o,\tau_o) = (\sigma,\tau)$, then indeed τ is a substring of λ. If $(\sigma_o,\tau_o) \neq (\sigma,\tau)$, then, by 2.1.2, τ_o is disjoint from τ, and (σ,τ) is a

production of T_o. By 2.1.1, since every production of T_o is a production of T, every node of τ is a terminal node of T_o. By inductive hypothesis then, τ is a substring of $\lambda_1\sigma_o\lambda_2$. τ is disjoint from σ_o by 2.1.1. Hence τ is a substring either of λ_1 or of λ_2. In either case τ is a substring of λ.

To construct T' now, let us assume τ is a substring of λ_1 and observe as we go that the proof is easily adaptable to λ_2. So let $\lambda_1 = \lambda_1'\tau\lambda_1''$. By inductive hypothesis there exists a graph T_o' with terminal string $\lambda_1'\sigma\lambda_1''\sigma_o\lambda_2$ such that $T_o = T_o' + (\sigma,\tau)$. We let $T' = T_o' + (\sigma_o,\tau_o)$ according to PG2, and note that T' has terminal string $\lambda_1'\sigma\lambda_1''\tau_o\lambda_2$. T_o, T_o', and T' were all derived according to PG2, so they all have the same initial string as T. T' lacks only one production of T, namely (σ,τ). $T' + (\sigma,\tau)$ is a production graph according to PG2, and has the same productions and initial string as T. That is, $T' + (\sigma,\tau) = T$. The terminal string obtained from this construction is $\lambda_1'\tau\lambda_1''\tau_o\lambda_2 = \lambda_1\tau_o\lambda_2 = \lambda$. So the terminal string $\lambda_1'\sigma\lambda_1''\tau_o\lambda_2$ is the one required.

The uniqueness of T' is clear, for a production graph is completely specified by its productions and initial string.

PROPOSITION 2.1.4. A production graph has just one terminal string.

<u>Proof</u>. The assertion holds for graphs with no productions. Assume it holds for graphs with n productions and let T be a graph with n + 1 productions. Let λ_o be a terminal string of T. Then there is a graph T_o with terminal string $\rho_1\sigma_o\rho_2$, and a production (σ_o,τ_o) of T, such that $T = T_o + (\sigma_o,\tau_o)$ and $\lambda_o = \rho_1\tau_o\rho_2$. Similarly, if we let λ_1 be any other terminal string of T we have $T = T_1 + (\sigma_1,\tau_1)$, where T_1 has terminal string $\rho_1'\sigma_1\rho_2'$ and $\lambda_1 = \rho_1'\tau_1\rho_2'$.

By 2.1.3, τ_1 is a substring of $\rho_1\tau_o\rho_2$. If $(\sigma_o,\tau_o) = (\sigma_1,\tau_1)$ then $T_o = T_1$, for T_o and T_1 have the same initial string as T and would then have the same productions. By inductive hypothesis we could then say $\rho_1\sigma_o\rho_2 = \rho_1'\sigma_1\rho_2'$, whence, since $\sigma_o = \sigma_1$, $\rho_1 = \rho_1'$ and $\rho_2 = \rho_2'$. Thus, since $\tau_o = \tau_1$, $\rho_1\tau_o\rho_2 = \rho_1'\tau_1\rho_2'$, or $\lambda_o = \lambda_1$.

If $(\sigma_o,\tau_o) \neq (\sigma_1,\tau_1)$, then τ_o and τ_1 are disjoint according to 2.1.2. So τ_1 is a substring of either ρ_1 or ρ_2. There is no loss of generality in choosing ρ_1, so that $\rho_1 = \mu_1\tau_1\mu_2$. Then $\lambda_o = \mu_1\tau_1\mu_2\tau_o\rho_2$. By 2.1.3 there exists a production graph T' with terminal string $\mu_1\sigma_1\mu_2\tau_o\rho_2$ such that $T = T' + (\sigma_1,\tau_1)$. But T' must therefore have the same productions and initial string as T_1, that is, $T' = T_1$. By inductive hypothesis $\mu_1\sigma_1\mu_2\tau_o\rho_2 = \rho_1'\sigma_1\rho_2'$. Therefore $\mu_1 = \rho_1'$ and $\mu_2\tau_o\rho_2 = \rho_2'$. Now

$$\lambda_1 = \rho_1'\tau_1\rho_2' = \mu_1\tau_1\mu_2\tau_o\rho_2 = \rho_1\tau_o\rho_2 = \lambda_o.$$

We have thus taken two arbitrary terminal strings of T and found that they are the same.

The duality principle established next (2.1.6) will be used quite frequently.

PROPOSITION 2.1.5. If $T = (\Phi,\mu_1\mu_2\mu_3)$ is a production graph and σ is a string with no nodes in common with T, then $(\Phi \cup \{(\sigma,\mu_2)\},\mu_1\sigma\mu_3)$ is a production graph with the same terminal string as T.

<u>Notation</u>. $(\Phi \cup \{(\sigma,\mu_2)\},\mu_1\sigma\mu_3)$ will be denoted by $(\sigma,\mu_2) + T$.

Proof. If Φ is empty, we get the assertion by applying PG2 to $(\emptyset, \mu_1\sigma\mu_3)$. For the induction step on $T + (\tau, \rho)$, we form $(\sigma, \mu_2) + T$ by inductive hypothesis, note that it has the same terminal string as T and has the initial string of the graph we want to construct, and then form $((\sigma, \mu_2) + T) + (\tau, \rho)$ by PG2.

PROPOSITION 2.1.6. (Duality.) Let (Φ, μ) be a production graph with terminal string τ. Let $\Phi' = \{(\rho_2, \rho_1) : (\rho_1, \rho_2) \text{ in } \Phi\}$. Then (Φ', τ) is a production graph with terminal string μ.

Notation. (Φ', τ) will be called the dual of (Φ, μ).

Proof. If Φ is empty, then $\mu = \tau$ and the assertion holds. For the induction step on $T + (\rho_1, \rho_2)$, form the dual of T by inductive hypothesis, note that its initial string is the same as the terminal string of T, so that (ρ_2, ρ_1) can be adjoined according to 2.1.5.

The duality principle 2.1.6 gives us dual propositions to 2.1.1, 2.1.2, 2.1.3. For example, the dual of 2.1.1 asserts that a node is an initial node if and only if it does not occur in the second string of any production. The dual of 2.1.4 asserts that any two production graphs with the same productions and the same terminal string must have the same initial string, i.e., must be equal.

The next objective is to formulate some rules for making changes internally in a graph so that the resulting entity still is a production graph. These changes will be expressed in terms of the removal and replacement of "subgraphs."

By an _initial subgraph_ of a production graph $T = (\Phi, \mu)$, we mean a production graph (Φ', μ) such that $\Phi' \subset \Phi$. An _intermediate string_ of T is a terminal string of an initial subgraph of T. A _string_ of T is a substring of an intermediate string. By the terms _initial_ or _terminal_ substring of T, we mean respectively a substring of the initial or terminal string of T.

Notice that if T_1 is an initial subgraph of T_2 and T_2 is an initial subgraph of T_3, then T_1 is an initial subgraph of T_3. Consequently, if T_1 is an initial subgraph of T_2, then any intermediate string of T_1 is an intermediate string of T_2.

We call a string, production, or graph _disjoint_ from another string, production, or graph if the two entities have no nodes in common.

PROPOSITION 2.1.7. Let T_1 be an initial subgraph of a production graph T, let σ be a terminal substring of T_1, and let (σ, τ) be a production of T. Then τ is disjoint from T_1 and $T_1 + (\sigma, \tau)$ is an initial subgraph of T.

Proof. We have only to prove that no node of τ is a node of T_1, for then, by PG2, $T_1 + (\sigma, \tau)$ is a production graph with the same initial string as T_1, hence as T, and all of the productions of $T_1 + (\sigma, \tau)$ belong to T.

(σ, τ) is not a production of T_1, by 2.1.1. If (σ_1, τ_1) is any production of T_1, then, by 2.1.2 applied to T, τ_1 and τ are disjoint. Hence by the dual of 2.1.1, τ can contain, at most, initial nodes of T_1. But the initial nodes of T_1 are just the initial nodes of T, and none of these can be in τ by the dual of 2.1.1.

PROPOSITION 2.1.8. Let T_1 be an initial subgraph of a production graph T, with $T_1 \neq T$. There exists a terminal substring σ of T_1 and a production (σ,τ) of T such that $T_1 + (\sigma,\tau)$ is an initial subgraph of T.

Proof. By induction on the number of productions of T. If T has no productions, then T has no proper initial subgraph, so the assertion is true vacuously.

Suppose the assertion holds for graphs with n productions, and let $T = T' + (\sigma,\tau)$ have n + 1 productions. Let T_1 be an initial subgraph of T. If (σ,τ) is not a production of T_1, then T_1 is an initial subgraph of T'. If $T_1 = T'$, then the required terminal substring and production are σ and (σ,τ). If $T' \neq T_1$, then the required production is obtained from the inductive hypothesis.

Suppose (σ,τ) is a production of T_1. Then it is a terminal production of T_1, by 2.1.1. $T_1 = T_2 + (\sigma,\tau)$ for some T_2, by 2.1.3. T_2 is a proper initial subgraph of T' lest T_1 be equal to T. By inductive hypothesis there is a terminal substring σ_1 of T_2 and a production (σ_1,τ_1) of T' such that $T_2 + (\sigma_1,\tau_1)$ is an initial subgraph of T'. σ_1 is disjoint from σ, for otherwise we would have, by the dual of 2.1.2, that $(\sigma_1,\tau_1) = (\sigma,\tau)$, but (σ_1,τ_1) is in T' and (σ,τ) is not. Therefore σ_1 is a terminal substring of $T_2 + (\sigma,\tau) = T_1$. By 2.1.7, $T_1 + (\sigma_1,\tau_1)$ is an initial subgraph of T.

PROPOSITION 2.1.9. Let T be a production graph and σ a non-null string. If $\sigma_1\sigma$ and $\sigma\sigma_2$ are both strings of T, then $\sigma_1\sigma\sigma_2$ is a string of T.

Proof. It will suffice to consider just the case where σ has length 1, for the general case can be reduced to this by considering the rightmost node q of σ and the two strings $\sigma_1\sigma$ and $q\sigma_2$.

So we will consider strings $\sigma_1 p$ and $p\sigma_2$, with p a node of T. There exists an initial subgraph T_1 of T with n_1 productions, and another, T_2, with n_2 productions, containing $\sigma_1 p$ and $p\sigma_2$ respectively as terminal substrings. The proof is by induction on $n_1 + n_2$.

If $n_1 + n_2 = 0$, then $\sigma_1 p$ and $p\sigma_2$ are initial substrings of T, whence $\sigma_1 p\sigma_2$ is an initial substring of T, hence a string of T.

Suppose the assertion holds for all cases where $n_1 + n_2 \leq n$, and consider a case where $n_1 + n_2 = n + 1$.

Case 1. Suppose T_1 contains a terminal production (τ_1, τ_2) where τ_2 does not contain p. If τ_2 is also disjoint from σ_1, then $T_1 - (\tau_1, \tau_2)$ has terminal substring $\sigma_1 p$ and only $n_1 - 1$ productions, so that the inductive hypothesis is directly applicable. Suppose τ_2 is not disjoint from σ_1. Then, since p is not in τ_2, T_1 has a terminal substring $\rho_1\tau_2\rho_2 p$ where σ_1 is a rightmost substring of $\rho_1\tau_2\rho_2$. Hence $T_1 - (\tau_1, \tau_2)$ has a terminal substring $\rho_1\tau_1\rho_2 p$. The inductive hypothesis tells us that $\rho_1\tau_1\rho_2 p\sigma_2$ is a string of T. Let T_o be an initial subgraph of T containing $\rho_1\tau_1\rho_2 p\sigma_2$ as a terminal substring. By 2.1.7, $T_o + (\tau_1, \tau_2)$ is an initial subgraph of T. $T_o + (\tau_1, \tau_2)$ contains $\rho_1\tau_2\rho_2 p\sigma_2$. σ_1 being a rightmost substring of $\rho_1\tau_2\rho_2$, $\sigma_1 p\sigma_2$ must be a string of T.

Case 2. Suppose T_2 contains a terminal production (τ_1', τ_2') where τ_2' does not contain p. The argument of Case 1 is adaptable by symmetry to this case.

Now either T_1 or T_2 has at least one production, and by the symmetry of the situation we may as well assume that T_1 has a production. Then T_1 has a terminal production (τ_1,τ_2). By Case 1, we assume that τ_2 contains p.

By the dual of 2.1.1, p is not an initial node of T, hence not an initial node of T_2. By the dual of 2.1.1 and by 2.1.2, we learn that (τ_1,τ_2) is a production of T_2. Suppose (τ_1,τ_2) is not a terminal production of T_2. Then, since T_2 has a terminal production, by 2.1.2, T_2 must have a terminal production (τ_1',τ_2') where τ_2' does not contain p. But this is Case 2. So we can assume (τ_1,τ_2) is terminal in T_2.

Now if τ_2 contains all of σ_1, then $\sigma_1 p$ is a substring of τ_2, hence a terminal substring of T_2. This would imply that $\sigma_1 p \sigma_2$ is a terminal substring of T_2, i.e., that $\sigma_1 p \sigma_2$ is a string of T. So we can assume that τ_2 does not contain all of σ_1. Similarly, we can assume that τ_2 does not contain all of σ_2.

We conclude that there must exist strings σ_1' and σ_2' such that $\sigma_1'\tau_2\sigma_2' = \sigma_1 p \sigma_2$, with $\sigma_1'\tau_2$ a terminal substring of T_1 and $\tau_2\sigma_2'$ a terminal substring of T_2. Therefore $\sigma_1'\tau_1$ is a terminal substring of $T_1 - (\tau_1,\tau_2)$. By inductive hypothesis, $\sigma_1'\tau_1\sigma_2'$ is a string of T, hence occurs as a terminal substring of some initial subgraph T_0 of T. $T_0 + (\tau_1,\tau_2)$ is an initial subgraph of T, by 2.1.7, and contains the terminal substring $\sigma_1'\tau_2\sigma_2' = \sigma_1 p \sigma_2$, i.e., $\sigma_1 p \sigma_2$ is a string of T. This finishes the proof.

PROPOSITION 2.1.10. If (Φ,μ) is a production graph with terminal string τ, and if μ_1 and μ_2 are strings disjoint from (Φ,μ), then $(\Phi,\mu_1\mu\mu_2)$ is a production graph with terminal string $\mu_1\tau\mu_2$. Conversely, if $(\Phi,\mu_1\mu\mu_2)$ is a production

graph with terminal string $\mu_1\tau\mu_2$, then (Φ,μ) is a production graph with terminal string τ.

<u>Notation</u>. If T denotes (Φ,μ), then $(\Phi,\mu_1\mu\mu_2)$ will be denoted by $\mu_1 T\mu_2$.

<u>Proof</u>. The conclusion is direct from PG1 if Φ is empty. If the assertion holds for graphs with n productions and T has n + 1 productions, choose a terminal production (σ_1,σ_2) of T and form $\mu_1(T - (\sigma_1,\sigma_2))\mu_2 + (\sigma_1,\sigma_2)$. This is the required graph.

For the converse, the assertion is trivial if Φ is empty. Suppose the converse holds for all graphs with n productions and let $T + (\sigma_1,\sigma_2)$ have n + 1 productions, where $T + (\sigma_1,\sigma_2)$ has initial string $\mu_1\mu\mu_2$ and terminal string $\mu_1\tau\mu_2$. By the dual of 2.1.1, σ_2 contains no node of the initial substrings μ_1,μ_2. Clearly, therefore, T has terminal string $\mu_1\tau'\mu_2$ for some τ'. By inductive hypothesis, $T = \mu_1 U\mu_2$ for some graph U. Clearly σ_1 must be a terminal substring of U, hence $T + (\sigma_1,\sigma_2) = \mu_1(U + (\sigma_1,\sigma_2))\mu_2$. $U + (\sigma_1,\sigma_2)$ is the required graph.

PROPOSITION 2.1.11. Let (Φ,μ) be a production graph with terminal string $\tau_1\tau\tau_2$. Let (Φ_1,τ) be a production graph with terminal string σ, and with only the nodes of τ in common with (Φ,μ). Then $(\Phi \cup \Phi_1,\mu)$ is a production graph with terminal string $\tau_1\sigma\tau_2$.

<u>Notation</u>. If $(\Phi,\mu) = T$ and $(\Phi_1,\tau) = T_1$, then $(\Phi \cup \Phi_1,\mu)$ will be denoted by $T + T_1$ or by $T_1 + T$.

Proof. By induction on the size of Φ_1. The case where Φ_1 is empty is trivial. To form $T + (T_1 + (\rho_1, \rho_2))$ in the induction step, merely note that ρ_1 is a terminal substring of $T + T_1$, ρ_2 is disjoint from it, hence $(T + T_1) + (\rho_1, \rho_2)$ is a production graph and has the properties required for $T + (T_1 + (\rho_1, \rho_2))$.

Let us now define a subgraph of a production graph $T = (\Phi, \mu)$ to be a production graph (Φ', σ) such that $\Phi' \subset \Phi$ and σ is a string of T.

Note that it is essential to have the initial string be a string of T, that is, occur in the terminal string of some initial subgraph. For, the idea of a subgraph is that it can be viewed as an integral component of T. It can happen that T contains a set of productions which form a production graph but not a subgraph of T.

The next proposition generalizes 2.1.7, with a subgraph in the role of the production.

PROPOSITION 2.1.12. Let T_1 be an initial subgraph and T_2 a subgraph of the production graph T. If the initial string of T_2 is terminal substring of T_1, then T_1 and T_2 have no nodes in common other than the initial nodes of T_2. Thus $T_1 + T_2$ exists and is an initial subgraph of T.

Proof. By induction on the number of productions in T_2. The assertion is trivial when T_2 has no productions. Suppose the assertion holds in all cases where T_2 has n productions and consider a case where T_2 has n + 1 productions. Then $T_2 = T_2' + (\sigma, \tau)$ for some terminal production (σ, τ) of T_2. σ is a terminal substring of T_2'. T_2' has the same initial string as T_2, and has only n productions. Therefore T_2' has no nodes in common with T_1 other than initial

nodes, and $T_1 + T_2'$ is an initial subgraph of T. σ is a terminal substring of $T_1 + T_2'$, by 2.1.11. So, by 2.1.7, τ is disjoint from $T_1 + T_2'$ and $(T_1 + T_2') + (\sigma,\tau)$ is an initial subgraph of T. The productions and initial word of $T_1 + (T_2' + (\sigma,\tau))$ are the same as those of $(T_1 + T_2') + (\sigma,\tau)$, so the induction is continued to n + 1.

PROPOSITION 2.1.13. Let (Φ,μ) be a production graph with terminal string $\tau_1\tau\tau_2$. If (Φ_1,σ) is a subgraph of (Φ,μ) with terminal string τ, then $(\Phi - \Phi_1,\mu)$ is a production graph with terminal string $\tau_1\sigma\tau_2$.

Notation. If $T = (\Phi,\mu)$ and $T_1 = (\Phi_1,\sigma)$, then $(\Phi - \Phi_1,\mu)$ will be denoted by $T - T_1$. Also, in the case where the initial string σ of T_1 is a substring of the initial string $\mu_1\sigma\mu_2$ of T, with no hypothesis on the terminal string τ of T_1, the graph $(\Phi - \Phi_1,\mu_1\tau\mu_2)$, which is obtained from the dual of 2.1.13, will also be denoted by $T - T_1$.

Proof. By induction on the size of Φ_1. If Φ_1 is empty, then the assertion is trivial. Suppose the assertion holds in all cases where $T_1 = (\Phi_1,\sigma)$ has n productions, and consider a case where T_1 has n + 1 productions. Let (ρ_1,ρ_2) be a terminal production of T_1. Let $T = (\Phi,\mu)$. To assert that the required production graph is $(T - (\rho_1,\rho_2)) - (T_1 - (\rho_1,\rho_2))$ we need only show that $T_1 - (\rho_1,\rho_2)$ is a subgraph of $T - (\rho_1,\rho_2)$, for clearly the terminal string of $T_1 - (\rho_1,\rho_2)$ is a terminal substring of $T - (\rho_1,\rho_2)$. The initial string σ of T_1 is also the initial string of $T_1 - (\rho_1,\rho_2)$. Let T_o be an initial subgraph of T containing σ as a terminal substring. T_o could not contain (ρ_1,ρ_2) because of 2.1.12. Thus every production of T_o is a production of $T - (\rho_1,\rho_2)$,

and T_0 has the same initial string as T and $T - (\rho_1, \rho_2)$, so T_0 is an initial subgraph of $T - (\rho_1, \rho_2)$. Hence σ is a string of $T - (\rho_1, \rho_2)$, so T_1 is a subgraph of $T - (\rho_1, \rho_2)$.

COROLLARY. Let $T' = (\Phi', \tau)$ be the dual of the production graph $T = (\Phi, \mu)$. Let $T_0 = (\Phi_0, \mu)$ be an initial subgraph of T, let $T_0' = (\Phi_0', \sigma)$ be the dual of T_0. Then $T' - T_0'$ is an initial subgraph of T' with terminal string σ.

PROPOSITION 2.1.15. Let $T = (\Phi, \mu)$ be a production graph, let $\Phi_1 \subset \Phi$. $T_1 = (\Phi_1, \mu_1)$ is a subgraph of T if and only if the terminal string of T_1 is a string of T and T_1 is a production graph.

<u>Proof</u>. Suppose T_1 is a subgraph of T. Then μ_1 is a string of T, i.e., μ_1 is a terminal substring of an initial subgraph T_0 of T. By 2.1.12, $T_0 + T_1$ is an initial subgraph of T. Therefore the terminal string of T_1 is a string of T.

Conversely, suppose the terminal string of T_1 is a string of T. Then the dual of T_1 is a subgraph of the dual of T, for, by 2.1.13 applied to the dual of T and the dual of an initial subgraph T_0 of T, we see that the dual of $T - T_0$ is an initial subgraph of the dual of T and has the same terminal string as T_0. Thus the strings of T are just the strings of the dual of T. So the initial string of the dual of T_1 is a string of the dual of T. By the first part of the assertion being proved, the terminal string of the dual of T_1, i.e., the initial string of T_1, is a string of the dual of T, hence a string of T. Therefore T_1 is a subgraph of T.

COROLLARY. If T_1 is a subgraph of T then the dual of T_1 is a subgraph of the dual of T.

Let $T = (\Phi,\mu)$ be a production graph. Let T_1 be a subgraph of T. Let T_2 be a production graph with the same initial and terminal strings, respectively, as T_1, but with no other nodes in common with T. Then we say that T_1 is <u>replaceable</u> <u>by</u> T_2 <u>in</u> T.

PROPOSITION 2.1.16. Let $T = (\Phi,\mu)$ be a production graph with terminal string τ. Let $T_1 = (\Phi_1,\mu_1)$ be replaceable by $T_2 = (\Phi_2,\mu_2)$ in T. Then $((\Phi - \Phi_1) \cup \Phi_2,\mu)$ is a production graph with terminal string τ.

<u>Notation</u>. $((\Phi - \Phi_1) \cup \Phi_2,\mu)$ will be called <u>the</u> <u>graph</u> <u>formed</u> <u>by</u> <u>replacing</u> T_1 <u>by</u> T_2 <u>in</u> T. We will use some obvious grammatical derivatives of this.

<u>Proof</u>. Let T_o be an initial subgraph of T with terminal string $\mu_o\mu_1\mu_o'$. By 2.1.12 and 2.1.11, $T_o + T_1$ is an initial subgraph of T with terminal string $\mu_o\tau_1\mu_o'$. By the dual of 2.1.13, $T - (T_o + T_1)$ is a subgraph of T with initial string $\mu_o\tau_1\mu_o'$ and with terminal string τ. By 2.1.11, $T_o + T_2$ is a production graph with initial string μ and terminal string $\mu_o\tau_1\mu_o'$. We want to form $(T_o + T_2) + (T - (T_o + T_1))$. We can do this by 2.1.11, provided that the two graphs have no nodes in common other than nodes of $\mu_o\tau_1\mu_o'$. By 2.1.12, $T_o + T_1$ has, except for nodes of $\mu_o\tau_1\mu_o'$, no nodes in common with $T - (T_o + T_1)$. Except for nodes of μ_1 and τ_1, T_2 has no nodes in common with T, hence none in common with $T - (T_o + T_1)$. So if $T_o + T_2$ has a nonterminal node p belonging also to $T - (T_o + T_1)$, then p must be in μ_1 and not in τ_1. Therefore, by 2.1.1 applied

to T_1, p must occur in ρ_1 for some production (ρ_1, ρ_2) of T_1. But then p is, by 2.1.1, a nonterminal node of $T_o + T_1$ and hence cannot belong to $T - (T_o + T_1)$. So the two graphs have the required property and their sum can be formed.

All additions are of the type that preserves the initial string of the term on the left, so $(T_o + T_2) + (T - (T_o + T_1))$ has initial string μ. Letting Φ_o denote the set of productions of T_o, the productions of $(T_o + T_2) +$ $(T - (T_o + T_1))$ are

$$(\Phi_o \cup \Phi_2) \cup (\Phi - (\Phi_o \cup \Phi_1)) = (\Phi_o \cup \Phi_2) \cup (\Phi - \Phi_1)$$
$$= \Phi_2 \cup (\Phi - \Phi_1).$$

The last equation holds because, if $\Phi_o \cap \Phi_1$ were not empty, then some production (ρ_1, ρ_2) would be common to T_o and T_1, whence the nodes of ρ_2 would be noninitial in T_1, yet belong to T_o, and we agreed that this does not happen.

$(T_o + T_2) + (T - (T_o + T_1))$ thus has the required initial word and the required set of productions.

PROPOSITION 2.1.17. Let (Φ_1, μ) and (Φ_2, μ) be initial subgraphs of the production graph (Φ, μ). Then $(\Phi_1 \cap \Phi_2, \mu)$ is an initial subgraph of (Φ, μ). Moreover, if σ is a terminal substring of Φ_1, and if all the nodes of σ occur in productions of Φ_2, then σ is a terminal substring of $(\Phi_1 \cap \Phi_2, \mu)$.

Notation. $(\Phi_1 \cap \Phi_2, \mu)$ will be denoted by $(\Phi_1, \mu) \cap (\Phi_2, \mu)$.

Proof. By induction on the number of productions of Φ_1. If Φ_1 is empty, then $(\Phi_1 \cap \Phi_2, \mu) = (\Phi_1, \mu)$, an initial subgraph of (Φ, μ). Moreover, the terminal string of $(\Phi_1 \cap \Phi_2, \mu)$ is in this case μ, so that any terminal substring is preserved in the intersection.

Suppose the assertion holds in all cases where Φ_1 has n productions, and consider a case where Φ_1 has n + 1 productions. $(\Phi_1,\mu) = (\Phi_1',\mu) + (\rho_1,\rho_2)$ for some terminal production (ρ_1,ρ_2) of (Φ_1,μ).

Suppose (ρ_1,ρ_2) is not in Φ_2. Then $\Phi_1 \cap \Phi_2 = \Phi_1' \cap \Phi_2$, whence $(\Phi_1 \cap \Phi_2,\mu)$ is an initial subgraph by inductive hypothesis. Moreover, suppose σ is a terminal substring of (Φ_1,μ) such that all nodes of σ are in (Φ_2,μ). Now σ is disjoint from ρ_2, for, by the dual of 2.1.1, no node of ρ_2 is in μ, hence any node of ρ_2 in (Φ_2,μ) is a noninitial node of (Φ_2,μ), hence occurs in the second string of some production of (Φ_2,μ), and, by the dual of 2.1.2, this could only be the production (ρ_1,ρ_2). But we are assuming that (ρ_1,ρ_2) is not in (Φ_2,μ), so we conclude that σ is disjoint from ρ_2. It follows that σ is a terminal substring of (Φ_1',μ) and hence, by inductive hypothesis, a terminal substring of $(\Phi_1 \cap \Phi_2,\mu)$.

Suppose (ρ_1,ρ_2) is in Φ_2, ρ_1 is a terminal substring of (Φ_1',μ), and all the nodes of ρ_1 are in (Φ_2,μ). By inductive hypothesis $(\Phi_1' \cap \Phi_2,\mu)$ is an initial subgraph with terminal substring ρ_1. Hence $(\Phi_1' \cap \Phi_2,\mu) + (\rho_1,\rho_2)$ is an initial subgraph, and its productions are $(\Phi_1' \cap \Phi_2) \cup \{(\rho_1,\rho_2)\} = \Phi_1 \cap \Phi_2$. Thus $(\Phi_1 \cap \Phi_2,\mu)$ is an initial subgraph.

Moreover, suppose σ is a terminal substring of (Φ_1,μ) such that every node of σ is in (Φ_2,μ). If σ is disjoint from ρ_2 then (Φ_1',μ) contains σ as a terminal substring and σ is disjoint from ρ_1. By inductive hypothesis $(\Phi_1' \cap \Phi_2,\mu)$ contains both σ and ρ_1 as terminal substrings. σ and ρ_1 being disjoint, $(\Phi_1' \cap \Phi_2,\mu) + (\rho_1,\rho_2) = (\Phi_1 \cap \Phi_2,\mu)$ contains σ as a terminal substring. If σ is not disjoint from ρ_2, then there exists a terminal substring $\sigma_1\rho_2\sigma_2$

of (Φ_1,μ) containing σ as substring, and where every node of $\sigma_1\rho_2\sigma_2$ is in (Φ_2,μ). $\sigma_1\rho_1\sigma_2$ is a terminal substring of (Φ_1',μ), hence of $(\Phi_1' \cap \Phi_2,\mu)$ by inductive hypothesis. $\sigma_1\rho_2\sigma_2$ is therefore a terminal substring of $(\Phi_1' \cap \Phi_2,\mu) + (\rho_1,\rho_2) = (\Phi_1 \cap \Phi_2,\mu)$. Therefore σ is a terminal substring of $(\Phi_1 \cap \Phi_2,\mu)$.

PROPOSITION 2.1.18. For each production (σ,τ) of a production graph T, σ and τ are strings of T.

Proof. By induction on the number of productions in T. The assertion is vacuously true when T has no productions. Suppose the assertion is true in all cases where T has n productions, and consider a case where T has n + 1 productions. Then $T = T' + (\sigma',\tau')$ where T' has n productions. If (σ,τ) is in T', then σ and τ are strings of T' by inductive hypothesis, hence strings of T. Otherwise $(\sigma,\tau) = (\sigma',\tau')$. σ' is a string of T because it is a terminal substring of the initial subgraph T' of T. τ is a string of T because it is a terminal substring of T.

COROLLARY. Every node of T is a string of T.

Proof. Every initial node is in the initial subgraph $(\emptyset,\mu)$; every terminal node is terminal in the initial subgraph T. All other nodes occur in productions, by 2.1.1 and its dual, hence are strings of T by 2.1.18.

PROPOSITION 2.1.19. Let T_o be an initial subgraph of a production graph T and let T_1 be a subgraph of T. If T_o contains all the noninitial terminal nodes of T_1, then T_o contains all the productions of T_1.

Proof. By induction on the number of productions in T_1. If T_1 has no productions, then we have nothing to prove. Let T_1 have n + 1 productions where the assertion holds for all subgraphs having n productions. Let $T_1 = T_1' + (\sigma, \tau)$. The nodes of τ are terminal and noninitial in T_1 and therefore are all in T_o. By 2.1.7, (σ, τ) is in T_o. Thus the nodes of σ are in T_o. A noninitial terminal node of T_1' is either in σ or is a noninitial terminal node of T_1. So by inductive hypothesis T_o contains all the productions of T_1'. Since it also contains (σ, τ), it contains all the productions of T_1.

PROPOSITION 2.1.20. Let T_o be an initial subgraph of a production graph T, let T_1 be a subgraph of T. If T_o contains all the terminal nodes of T_1, then T_1 is a subgraph of T_o.

Proof. We only have to show that the terminal string τ of T_1 is a string of T_o. There exists an initial subgraph U of T containing τ as terminal substring. By 2.1.17, $U \cap T_o$ is an initial subgraph of T containing τ as terminal substring. $U \cap T_o$ is also an initial subgraph of T_o, therefore τ is a string of T_o.

COROLLARY. (By duality.) If $T - T_o$ contains all the initial nodes of T_1, then T_1 is a subgraph of $T - T_o$.

COROLLARY. (By duality.) If T_o fails to contain any of the nonterminal initial nodes of T_1, then T_o fails to contain any production of T_1.

While 2.1.16 allows us to replace a single subgraph, we will need to be able to replace more than one subgraph at the same time. It can happen that

after the replacement of one subgraph the next one is no longer present as a subgraph, even though all of its nodes and productions are still there. We next set forth some conditions under which such difficulties do not occur.

DEFINITION. Two subgraphs of a production graph T are separable in T if there exists an initial subgraph T_o of T which contains all the nodes of one while $T - T_o$ contains all the nodes of the other.

DEFINITION. Let T_1 and T_2 be subgraphs of a production graph T. Let U_1 and U_2 be production graphs. T_1, T_2 are replaceable by U_1, U_2 in T if T_1 and T_2 are separable, T_1 is replaceable by U_1 in T, T_2 is replaceable by U_2 in T, and U_1 and U_2 have no common nodes other than their initial or terminal nodes. $T_1, \ldots, T_k$, all subgraphs of T, are replaceable by $U_1, \ldots, U_k$ in T if for all $i \neq j$ it is true that T_i, T_j are replaceable by U_i, U_j in T.

PROPOSITION 2.1.21. Let T_1 and T_2 be subgraphs of a production graph T. Let T_1, T_2 be replaceable by U_1, U_2 in T. Let V be formed by replacing T_1 by U_1 in T. Then T_2 is a subgraph of V and T_2 is replaceable by U_2 in V.

Notation. The graph formed by replacing T_2 by U_2 in V will be referred to as the graph formed by replacing T_1, T_2 by U_1, U_2 in T. Moreover, if $T_1, \ldots, T_k$ are replaceable by $U_1, \ldots, U_k$ in T, then repeated use of 2.1.21 allows us to speak, with the obvious meaning, of the graph formed by replacing $T_1, \ldots, T_k$ by $U_1, \ldots, U_k$ in T. Clearly, if π is any permutation on the first k integers, then, in view of 2.1.21, the graph formed by replacing $T_{\pi(1)}, \ldots, T_{\pi(k)}$ with $U_{\pi(1)}, \ldots, U_{\pi(k)}$ in T is the same as the graph formed by replacing $T_1, \ldots, T_k$ by $U_1, \ldots, U_k$ in T.

<u>Proof</u>. <u>Case 1</u>. Suppose there exists an initial subgraph T_o of T containing all the nodes of T_1 while T - T_o contains all the nodes of T_2. By 2.1.20 and its dual, T_1 is a subgraph of T_o and T_2 is a subgraph of T - T_o. T_1 is replaceable by U_1 in T_o. Let U_o be the graph formed by replacing T_1 by U_1 in T_o. By merely checking productions and initial strings we see that U_o + (T - T_o) is the graph V formed by replacing T_1 by U_1 in T. T_2 is a subgraph of V, because it is a subgraph of T_1 - T_o. U_2 has no nodes in common with U_1 or with T except for initial or terminal nodes of U_2, hence no nodes in common with U_o or T - T_o except for initial or terminal nodes of U_2. U_2 still has the same initial and terminal strings as T_2. Therefore T_2 is replaceable by U_2 in V.

<u>Case 2</u>. There is an initial subgraph T_o of T which contains all the nodes of T_2 while T - T_o contains all the nodes of T_1. Looking at the duals of these graphs we see Case 1 again, and conclude that the dual of T_2 is replaceable by the dual of U_2 in the dual of V. Therefore T_2 is replaceable by U_2 in V.

PROPOSITION 2.1.22. Let T_1 and T_2 be two subgraphs with no productions in common. Suppose neither subgraph has a node which is both initial and terminal in the subgraph. If each of T_1,T_2 has just one initial production, or if each has just one terminal production, then T_1 and T_2 are separable.

<u>Proof</u>. Suppose each subgraph has just one initial production. Let the respective initial productions be (σ_1,τ_1) and (σ_2,τ_2). By 2.1.18, there exists an initial subgraph T_o containing τ_1 as a terminal substring. Suppose T_o

contains a node of τ_2. Then by 2.1.7, T_o contains (σ_2, τ_2). Since $(\sigma_1, \tau_1) \neq (\sigma_2, \tau_2)$, (σ_2, τ_2) is in the initial subgraph $T_o - (\sigma_1, \tau_1)$. By 2.1.18, there must exist an initial subgraph T_o' of T_o containing τ_2 as terminal substring. No node of τ_1 is in T_o. T_o' is an initial subgraph of the main graph, which we now name T. Thus by a change of notation if necessary we can assume that τ_1 is a terminal subword of an initial subgraph T_o which contains no nodes of τ_2.

No node of τ_2 is an initial node of $T - T_o$, for the initial nodes of $T - T_o$ are the terminal nodes of T_o. Hence, by 2.1.1 and 2.1.2, (σ_2, τ_2) is in $T - T_o$. By 2.1.19, since σ_2 must be the initial string of T_2, $T - T_o$ contains all the nodes of T_2.

τ_1 is a terminal substring of T_o. σ_1 is the initial string of T_1, hence τ_1 is the initial string of $T_1 - (\sigma_1, \tau_1)$. So $T_o + (T_1 - (\sigma_1, \tau_1)) = U$ is an initial subgraph of T. U contains no production of T_2, for T_o does not, and only productions of T_1 are added to form U. T-U therefore contains all the nodes of T_2, because every node of T_2 is in a production of T_2 lest some node of T_2 be both initial and terminal, so that should T-U lack a node of T_2 it must also lack a production of T_2. U contains all the nodes of T_1 for the same kind of reason. Therefore T_1 and T_2 are separable.

2.2. Labeled production graphs and ambiguity.

DEFINITION. A <u>labeled</u> production graph is a production graph in which each node is an ordered pair (p,X), where p is the <u>node name</u> and X is the <u>node label</u>.

Notation. It will seldom be necessary to make explicit reference to the node name. We will continue to use $p,q,r,\ldots$ as symbols for nodes.

For a node p, $\varphi(p)$ will denote the label of p.

For a string $\sigma = p_1 \ldots p_k$, the label of σ, written $\varphi(\sigma)$, is $\varphi(p_1)\ldots\varphi(p_k)$.

For a production (σ,τ) of a labeled production graph, the label of (σ,τ), written $\varphi((\sigma,\tau))$, is a production $\varphi(\sigma) \rightarrow \varphi(\tau)$.

Let T be a labeled production graph with initial string μ and terminal string τ. Then $\varphi(\mu)$ is called the initial word of T, and $\varphi(\tau)$ is called the terminal word of T. Continuing in the same vein, an initial (terminal) subword of T is a subword of the initial (terminal) word of T, an intermediate word of T is the label of an intermediate string of T, a word of T is the label of a string of T.

DEFINITION. Let G be a grammar. A labeled production graph T is a G-graph if, for every production (σ,τ) of T, $\varphi(\sigma) \rightarrow \varphi(\tau)$ is a production of G.

If G is a grammar with start letter S and external alphabet Σ, then a complete G-graph is a G-graph with initial word S and terminal word in Σ^*.

Let G be a grammar. Let us show that $\alpha \vdash_G \beta$ if and only if there exists a G-graph with initial word α and terminal word β.

In one direction we proceed by induction on the length of the derivation. For a derivation of length 1, the required G-graph is merely a sequence of labeled nodes with no productions. Suppose that for all derivations $\alpha_1,\ldots,\alpha_n$ of length n there is a corresponding G-graph with initial word α_1 and terminal word α_n, and consider a derivation $\alpha_1,\ldots,\alpha_{n+1}$. Let $\alpha_n = \beta_1\beta_2\beta_3$, $\alpha_{n+1} = \beta_1\gamma\beta_3$,

where $\beta_2 \to \gamma$ is a production of G. There is a graph T with initial string σ, and terminal string $\tau_1\tau_2\tau_3$, such that $\omega(\sigma) = \alpha_1$, $\omega(\tau_1) = \beta_1$, $\omega(\tau_2) = \beta_2$, $\omega(\tau_3) = \beta_3$. Choose a string ρ of new nodes so that $\omega(\rho) = \gamma$. $T + (\tau_2, \rho)$ has terminal string $\tau_1\rho\tau_3$, and terminal word $\omega(\tau_1)\omega(\rho)\omega(\tau_3) = \beta_1\gamma\beta_3 = \alpha_{n+1}$, and the initial word α_1.

In the other direction we proceed by induction on the number of productions in the G-graph. If the G-graph has no productions then it corresponds in the required way with a one-word derivation. Suppose a G-graph T has initial word α and terminal word β and that $\alpha \vdash_G \beta$. Suppose that T has terminal string $\tau_1\tau_2\tau_3$, and that $T + (\tau_2, \rho)$ is a G-graph. The terminal word of $T + (\tau_2, \rho)$ is $\omega(\tau_1)\omega(\rho)\omega(\tau_3)$.

Clearly $\beta = \omega(\tau_1)\omega(\tau_2)\omega(\tau_3) \vdash_G \omega(\tau_1)\omega(\rho)\omega(\tau_3)$, whence $\alpha \vdash_G \omega(\tau_1)\omega(\rho)\omega(\tau_3)$.

Notice that the construction of the G-graph from a given derivation, as set forth in the previous proof, is unique except for the choice of node names, given for each i the exact production which sends α_i to α_{i+1}. The derivation itself does not necessarily provide this information, but it does in the case of "leftmost" derivations of context free grammars, which we will be considering shortly.

Two graphs that are the same except for node names will be called <u>congruent</u>. Formally:

DEFINITION. Two labeled graphs, (Φ_1, μ_1) and (Φ_2, μ_2), are <u>congruent</u> if there exists a one-to-one function h from the nodes of (Φ_1, μ_1) onto the nodes of (Φ_2, μ_2) such that:

(i) $\mu_2 = h(\mu_1)$;

(ii) (σ, τ) is a production of Φ_1 if and only if $(h(\sigma), h(\tau))$ is a production of Φ_2;

(iii) $\varphi(p) = \varphi(h(p))$ for every node p of (Φ_1, μ_1).

By $h(p_1 \ldots p_k)$ we mean, of course, $h(p_1) \ldots h(p_k)$.

If P is a production of Φ_1, $P = (\sigma, \tau)$, then by $h(P)$ we mean the production $(h(\sigma), h(\tau))$ of Φ_2. If Φ' is a subset of Φ_1, then by $h(\Phi')$ we mean the subset $\{h(P) : P \text{ in } \Phi'\}$ of Φ_2.

It is easily seen that if (Φ', σ) is a subgraph (initial subgraph) of (Φ_1, μ_1), then $h((\Phi', \sigma)) = (h(\Phi'), h(\sigma))$ is a subgraph (initial subgraph) of (Φ_2, μ_2).

If the terminal string of (Φ_1, μ_1) is τ, then $h(\tau)$ is the terminal string of (Φ_2, μ_2). Conversely, we can conclude that the two graphs are congruent if they satisfy (ii) and (iii), and have respective terminal strings τ_1 and τ_2 such that $h(\tau_1) = \tau_2$. The latter assertion follows from the duality principle.

PROPOSITION 2.2.1. Condition (ii) in the definition of congruence can be weakened to:

(ii′) If (σ, τ) is a production of Φ_1, then $(h(\sigma), h(\tau))$ is a production of Φ_2.

Proof. Suppose two labeled graphs T_1 and T_2 satisfy (i), (ii′), and (iii) with respect to the one-to-one mapping h from the nodes of T_1 onto the nodes of T_2. Suppose $(h(\sigma), h(\tau))$ is a production of T_2 while (σ, τ) is not a production of T_1. No node of τ could be an initial node of T_1 because by (i) p

is an initial node only if $h(p)$ is, but, by the dual of 2.1.2, no node o[illegible]) is an initial node of T_2. Therefore, by the dual of 2.1.2, for any node p of τ there is a production (σ',τ') of T_1 where p is in τ'. $(h(\sigma'),h(\tau))$ is by (ii′) a production of T_2, and $h(p)$ is in $h(\tau')$. $(\sigma',\tau') \neq (\sigma,\tau)$, and since h is one-to-one, this entails $(h(\sigma'),h(\tau')) \neq (h(\sigma),h(\tau))$. But then $h(p)$ occurs in the second string of two productions of T_2, contrary to the dual of 2.1.2. This contradiction results from assuming that (ii) fails to hold.

PROPOSITION 2.2.2. There is at most one congruence between two given labeled production graphs.

Proof. Suppose h_1 and h_2 are congruences from T_1 to T_2. Let μ_1 and μ_2 be the initial strings of T_1 and T_2 respectively. Then $h_1(\mu_1) = h_2(\mu_1)$, i.e., the two agree on the initial nodes. In other words, h_1 and h_2 agree on the nodes of the initial subgraph of T_1, which contains no productions. Suppose they agree on the nodes of any initial subgraph of T_1 containing n productions, and let $T_0 + (\sigma,\tau)$ be an initial subgraph of T_1 containing $n + 1$ productions. Then $h_1(\sigma) = h_2(\sigma)$, because σ is a string of T_0 and T_0 only has n productions. $(h_1(\sigma),h_1(\tau))$ and $(h_2(\sigma),h_2(\tau))$ are productions of T_2. Lest the nodes of $h_1(\sigma) = h_2(\sigma)$ appear in the first strings of two distinct productions, we must have $h_1(\tau) = h_2(\tau)$. Hence h_1 and h_2 agree on all the nodes of $T_0 + (\sigma,\tau)$. This extends the assertion to all initial subgraphs of T_1, one of which is T_1 itself.

PROPOSITION 2.2.3. Let T and T' be congruent. Let $T_1,\ldots,T_k$ be replaceable by $U_1,\ldots,U_k$ in T, and suppose the congruence h from T to T' is extended to the

nodes of the U_i so that h is a congruence from U_i to $h(U_i)$ for $i = 1,\ldots,k$, and so that $h(T_1),\ldots,h(T_k)$ are replaceable by $h(U_1),\ldots,h(U_k)$ in T'. Then the graph formed by replacing $T_1,\ldots,T_k$ by $U_1,\ldots,U_k$ in T is congruent to the graph formed by replacing $h(T_1),\ldots,h(T_k)$ by $h(U_1),\ldots,h(U_k)$ in T'.

Proof. h is obviously an onto mapping. Let us see whether h is one-to-one. If p and q are both nodes of T, then $h(p) \neq h(q)$ unless $p = q$, for h is a congruence from T. If q is not a node of T, then q is a noninitial nonterminal node of one of the U_i. Hence $h(q)$ is a noninitial nonterminal node of $h(U_i)$. Because of the hypothesis about replaceability in T', $h(q)$ can belong neither to T' nor to any $h(U_j)$ other than $h(U_i)$. Therefore if $h(p) = h(q)$, p must not belong either to T or to any U_j other than U_i. p has to belong to something, so p must belong to U_i. Since h is a congruence on U_i, it follows that $p = q$. Therefore h is one-to-one.

If μ is the initial string of T, then $h(\mu)$ is the initial string of T', and the replacements do not change the respective initial strings. Thus (i) is satisfied.

Let (σ,τ) be a production of the graph V formed by the replacement of $T_1,\ldots,T_k$ by $U_1,\ldots,U_k$ in T. If (σ,τ) is in T, then (σ,τ) is in none of the T_i because all of the productions of the T_i have been replaced. Then $(h(\sigma),h(\tau))$ is in T' without being in any of the $h(T_i)$. Any production of T' not in any of the $h(T_i)$ must still be there after the replacement is made, so $(h(\sigma),h(\tau))$ is in the replacement graph. If (σ,τ) is not in T, then it is in some U_i, whence $(h(\sigma),h(\tau))$ is in $h(U_i)$. This again requires $(h(\sigma),h(\tau))$ to be in the replacement graph. Thus the assertion (ii′) of 2.2.1 holds.

(iii) holds for any node p either because p is in T or because p is one of the U_i.

Since we have (i), (ii$'$), and (iii), we have congruence by 2.2.1.

DEFINITION. A grammar G is unambiguous if every two complete G-graphs having the same terminal word are congruent.

Let us make sure that this notion of unambiguity reduces in the context free case to the usual notion.

The usual notion has been expressed in terms of trees which are easily seen to be just the context free case of our production graphs. However, the more familiar definition is in terms of leftmost generations [2].

Let G be a context free grammar. A derivation $\alpha_1,\ldots,\alpha_n$ is leftmost if for each $i < n$ we have $\alpha_i = \beta_i X_i \beta_i'$ and $\alpha_{i+1} = \beta_i \gamma_i \beta_i'$, where $X_i \to \gamma_i$ is a production of G and β_i contains only external letters. A context free grammar is unambiguous if for every word w which contains only external letters there exists at most one complete leftmost G-derivation ending with w.

To show that the two definitions of ambiguity are equivalent, we will establish a one-to-one correspondence between complete G-graphs and complete leftmost G-derivations. Given a complete G-graph T with initial word S and terminal word α_n, we will define a unique leftmost derivation $F(T) = \alpha_1,\ldots,\alpha_n$, where $\alpha_1 = S$, together with a corresponding sequence $F'(T) = T_1,\ldots,T_n$ of initial subgraphs of T such that α_i is the terminal word of each T_i and $T_n = T$. We will show that if $F(T) = F(T')$, then T is congruent to T'.

Let $T_1 = (\emptyset,p)$ if p is the initial string (necessarily of length 1) of T. $\alpha_1 = S$ is the terminal word of T_1. Suppose we have the initial subgraphs

$T_1,\ldots,T_j$ and a leftmost (partial) derivation $\alpha_1,\ldots,\alpha_j$ where each α_i is the terminal word of T_i. If $T_j = T$, then we are finished. Otherwise, let the terminal string of T_j be $\tau_1\tau_2$ where $\varphi(\tau_1)$ contains only external letters and $\varphi(\tau_2)$ begins with an internal letter. The terminal string of T_j must be in that form because there must be a production of T applicable to its terminal string, and we have adopted the convention that context free grammars do not have productions with external letters on the left. Let $\tau_2 = p\tau_2'$ where p is a node and $\varphi(p) = X$. Then $\alpha_j = \varphi(\tau_1)X\varphi(\tau_2')$. p is not a terminal node of T. Therefore T has a production (p,ρ). Let $T_{j+1} = T_j + (p,\rho)$. Let $\alpha_{j+1} = \varphi(\tau_1)\varphi(\rho)\varphi(\tau_2')$. α_{j+1} is the terminal word of T_{j+1}, and if $\alpha_1,\ldots,\alpha_j$ is a leftmost derivation, then so is $\alpha_1,\ldots,\alpha_{j+1}$. Eventually we must arrive at $T_n = T$ and a complete leftmost derivation $\alpha_1,\ldots,\alpha_n$. This completes the construction of F and F'. That $F(T) = F(T')$ implies T congruent to T' can be seen from the construction just set forth. Obviously T and T' would have to have the same number n of productions. Let $F'(T') = T_1',\ldots,T_n'$. Clearly T_1 is congruent to T_1'. Suppose T_j is congruent to T_j', under the congruence h. If $j = n$ we have the assertion. If not, then, using the notation of the construction, the terminal string of T_j' is $h(\tau_1)h(p)h(\tau_2')$. $\varphi(h(\tau_1))$ contains only external letters and $\varphi(h(p)) = X$. Clearly $T_{j+1}' = T_j' + (h(p),\rho')$ for some ρ'. ρ' has to have the same length as ρ since the length of α_{j+1} is fixed. $\varphi(\rho') = \varphi(\rho)$ in order to obtain α_{j+1} as terminal word of T_{j+1}'. The nodes of ρ and ρ' are foreign to T_j and T_j' respectively, so there is no difficulty about extending h so that $h(\rho) = \rho'$. Clearly the extended h is a congruence from T_{j+1} to T_{j+1}'.

We also need to know the converse, that if T is congruent to T' then $F(T) = F(T')$. It suffices to show that $F'(T)$ and $F'(T')$ are congruent term by term, for then their sequence of terminal words must be identical. T_1 and T_1' consist just of initial strings with the same label and are therefore congruent. Suppose T_j is congruent to T_j'. These are initial subgraphs of T and T', hence, there being at most one possible congruence between labeled graphs, the congruence between them must be a restriction of the congruence h from T to T'. $T_{j+1} = T_j + (p, \rho)$ where p is the leftmost terminal node of T_j such that $\varphi(p)$ is not an external letter. h(p) is, because of congruence, the leftmost terminal node of T_j' with label other than an external letter. In order that the sequence of terminal words be a leftmost derivation, $T_{j+1}' = T_j' + (h(p), \rho')$ for some ρ'. $(h(p), h(\rho))$ is a production of T', and h(p) cannot be in the first string of two distinct productions, so $\rho' = h(\rho)$. This clearly entails that T_{j+1} be congruent to T_{j+1}'.

It is clear that, given any leftmost derivation we can get a G-graph T such that F(T) is that leftmost derivation. This is clear by inspection of the general construction given earlier for obtaining G-graphs from G-derivations.

It follows now that if there are two different leftmost G-derivations of the same word, then there are two complete, incongruent G-graphs with the same terminal word. That is, if G is ambiguous in the leftmost derivation sense, then it is ambiguous in the production graph sense. Conversely, two incongruent complete G-graphs with the same terminal word must give rise via F to two distinct leftmost derivations of that word. That is, if G is ambiguous in the production graph sense, then it is ambiguous in the leftmost derivation sense.

Thus we have

PROPOSITION 2.2.4. A context free grammar is unambiguous in the sense of production graphs if and only if it is unambiguous in the sense of leftmost derivations.

It can be shown also that a context sensitive grammar is unambiguous in the sense of production graphs if and only if it is unambiguous in the sense of linear bounded automata, i.e., if and only if the language of the grammar is the language accepted by some linear bounded automaton which has at most one accepting computation for each word.

3. Context Limited Grammars

Let us now formulate, exactly, a definition of the context limited grammars, and then prove that they are equivalent, as described at the beginning, to the context free grammars.

Notation. _Lower than or equal to_ will denote a partial ordering relation, and _lower than_ will mean lower than or equal to and not equal to.

DEFINITION. A grammar G is _context limited_ if there exists a partial ordering _lower than or equal to_ on the alphabet of G such that for each production $\alpha \rightarrow \beta$ of G, every letter of α is lower than some letter of β.

Note, for example, that if X_1 is lower than Y_1, X_2 is lower than Y_2, and X_3 is lower than Y_3, then $X_1X_2X_3 \rightarrow Y_3Y_2Y_1$ could be a production. $XX \rightarrow X$ could not be a production.

If G is a context limited grammar then L(G) is called a _context limited language_. This is for present convenience only, since we will show that they are all context free.

Note that, as we have defined it, a context limited language cannot contain the null word ϵ. We could incorporate ϵ with a slight change in the definition, but instead, since it is generally agreed that the presence or absence of ϵ is of no fundamental importance, let us agree that the term "language" means a set without ϵ.

The context limited grammars are generalizations of the grammars, introduced by Ginsburg and Greibach [4], such that every production has an

external letter on the right. These grammars were proved in [4] to generate context free languages. Also, Ginsburg and Spanier have outlined an alternate proof that the context limited languages are context free, in which proof the Ginsburg-Greibach result just mentioned is applied. This proof could probably be generalized to incorporate the preservation of unambiguity, and possibly an easier proof than the present one could be so constructed. Note, however, that the present proof would be quite short without the ambiguity considerations.

Example 1. The language $\{a^n c b^p : p \text{ is the integer part of } \frac{n}{2}\}$ is generated by the context limited grammar with productions $\sigma \to a\sigma X$, $XX \to b$, $\sigma \to c$, $\sigma X \to c$.

Example 2. To illustrate the potential utility of the theorems we are about to prove, we use them to give a short proof of the known fact that if a sequential transducer M [3, p.91] is one-to-one on an unambiguous context free language L, then M(L) is an unambiguous context free language (a slight generalization of theorem 1 of [6], which is for generalized sequential machines). We assume without loss of generality that the input and output alphabets of M are disjoint. Let p_0 be the start state of M. Then p_0L is an unambiguous context free language. By Theorem 2 (below), p_0L is generated by an unambiguous context limited grammar G, the alphabet of which we can assume is disjoint from the output alphabet of M. We simply adjoin to G, productions corresponding to the moves of M, so that if M can go from $p\alpha$ to βq in one move, p and q being states, we adjoin a production $p\alpha \to \beta q$. We also adjoin $p\alpha \to \beta$ in order to make it possible to get rid of the state symbol at the end of the derivation: note that if $p\alpha \to \beta$ is used too soon, then a

complete derivation is not possible, because there will still exist internal letters but no production to handle them. The required partial ordering is obtained by extending that for G, so that every letter of G is lower than every output letter of M. Unambiguity follows from unambiguity of G and the one-to-one property of M.

The next Proposition is the easier direction of the main theorems.

PROPOSITION 3.1. If G is a context free grammar, then there exists a context limited grammar H such that L(H) = L(G), and if G is unambiguous then so is H.

Proof. The main result of [8] is that every context free grammar reduces, with unambiguity preserved, to one in which all productions are of the form

$$X \to a\alpha$$

where a is an external letter and X, of course, is not external. Under the partial ordering that makes every internal letter lower than every external letter, such a grammar is context limited.

Now let us reduce each context limited grammar to a context free one. We will need a few preliminary lemmas.

First let us recall the well-known fact that any partial ordering can be extended to a linear ordering.

PROPOSITION 3.2. If G is a context limited grammar, then G is context limited with respect to some linear ordering of its alphabet in which every internal letter is lower than every external letter.

<u>Proof</u>. We first obtain a linear ordering of the internal letters by extending the given partial ordering to a linear ordering, and then restricting the latter to the internal latters. Then we obtain a linear ordering of the external letters in the same way. Then we combine the two into a single linear ordering in which every internal letter is lower than every external letter.

Now there is no need to have the relation "a lower than X" for an external letter a, for by definition the external letters do not occur on the left in productions. So we can assume that the given partial ordering has this property. Then the linear ordering that we just constructed is an extension of the given partial ordering, hence G must be context limited with respect to it.

If G is a grammar, either context limited or context free, we call G <u>unambiguous</u> <u>modulo</u> the set R if whenever two complete G-graphs T_1 and T_2 have the same terminal word w, where w is in R, than T_1 is congruent to T_2.

We could prove directly that if a context limited grammar is unambiguous modulo a regular set R, then its language intersected with R is unambiguous. But this result will follow as a byproduct of the main result. We will, though, need to prove it in the special case where the grammar is context free.

PROPOSITION 3.3. If a context free grammar G is unambiguous modulo a regular set R, then $L(G) \cap R$ is an unambiguous context free language.

<u>Proof</u>. Let $A = (K, \Sigma, \delta, q_0, F)$ be a finite automaton (i.e., finite-state acceptor [3, p.47]) such that R is the set of words accepted by A.

We define a context free grammar H as follows. The alphabet of H is the external alphabet of G together with the triples (pXq), p and q states of A and X in the internal alphabet of G.

For two states p and q of A and a word α on the alphabet of H, call α (p,q)-connected if any of the following hold:

(i) α contains only input letters of A and $\delta(p,\alpha) = q$.

(ii) $\alpha = (pXq)$ for some X.

(iii) There are words $\alpha_1,\ldots,\alpha_k$ and states $p_1,\ldots,p_{k+1}$ such that $\alpha = \alpha_1\ldots\alpha_k$, and for each $i \le k$ α_i is (p_i,p_{i+1}) connected according to (i) and (ii).

Let S_H be the initial letter of H, where S_H is distinct from all letters mentioned so far. Let S be the initial letter of G. For each final state p of A, let H have the production

$$S_H \to (q_o Sp).$$

For a word α on the alphabet of H, let $\bar{\alpha}$ be the word on the alphabet of G obtained in the obvious way by replacing each (pXq) by X and leaving external letters unchanged.

For each internal letter (pXq) of H, and each (p,q)-connected word β such that $X \to \bar{\beta}$ is a production of G, let

$$(pXq) \to \beta$$

be a production of H.

It is obvious that if α is (p,q)-connected and $\alpha \mid\!\frac{}{H}\, \beta$, then β is (p,q)-connected.

Now if a word w on the external alphabet of A is (q_o,p)-connected where p is a final state, then w is in R.

From these two assertions, $L(H) \subset R$.

The productions of H being homomorphic to productions of G via the homomorphism which takes α to $\bar{\alpha}$, except for the productions involving S_H,

it is clear that if w has only external letters and $(q_0 Sp) \vdash_H w$, then $S \vdash_G w$. All complete H-derivations are of the form S_H, $(q_0 Sp), \ldots, w$. Therefore $L(H) \subset L(G)$. Therefore

$$L(H) \subset L(G) \cap R.$$

On the other hand, if any word α on the alphabet of H is (p,q)-connected, and if $X \vdash_G \bar{\alpha}$, then $(pXq) \vdash_H \alpha$. For, we have either $X = \bar{\alpha}$ or $X \to_G w_1 Y_1 \ldots w_k Y_k w_{k+1}$ where the w_i are words on the external alphabet and the Y_i are internal letters, with $Y_i \vdash_G \beta_i$ and $w_1 \beta_1 \ldots w_k \beta_k w_{k+1} = \alpha$. From this it is clear that we can prove the assertion by induction on the length of the derivation. In the induction step, we observe that if β_i is (p_i, p_{i+1})-connected then $(p_i Y_i p_{i+1}) \vdash_H \beta_i$. This together with $(pXq) \to_H w_1(p_1 Y_1 p_2) \ldots w_k(p_k Y_k p_{k+1}) w_{k+1}$ shows that $(pXq) \vdash_H \alpha$.

Now any word w in $R \cap L(G)$ is (q_0, p)-connected for some final state p. Therefore if $S \vdash_G w$, then $(q_0 Sp) \vdash_H \bar{w} = w$. This and $S_H \to_H (q_0 Sp)$ show that w is in L(H). Thus

$$R \cap L(G) \subset L(H).$$

Therefore

$$L(H) = L(G) \cap R.$$

The productions of H being homomorphic in the sense mentioned above to the productions of G, every complete H-graph is, ignoring the initial node, homomorphic in the same sense to a complete G-graph. That is, the complete G-graph is obtained by deleting the initial node and production and replacing each label (pXq) by X. Therefore if T_1 and T_2 are complete H-graphs, with the same terminal word, then there is a one-to-one function h from the noninitial

nodes of T_1 onto the noninitial nodes of T_2 such that $h(P)$ is a production of T_2 whenever P is a production of T_1, and such that for every noninitial node s of T_1, if $\varphi(s) = (pXq)$ then $\varphi(h(s)) = (p_1Xq_1)$, and if $\varphi(s) = a$ where a is an external letter then $\varphi(h(s)) = a$. In the former case we have that the terminal word of T_1 is $w_1w_2w_3$ where $(pXq) \vdash_{\overline{H}} w_2$, and w_1 is (q_o,p)-connected (with the convention that ϵ is (r,r)-connected for any r), and w_3 is (q,r)-connected for some final state r. The terminal word of T_2 is also $w_1w_2w_3$, and $(p_1Xq_1) \vdash_{\overline{H}} w_2$ (we rely, of course, on the unambiguity modulo R of G for the latter statement). Since w_2 has been ascertained from T_1 to be (p,q)-connected, and since A is deterministic, we must have $p_1 = p$ and $q_1 = q$. Therefore h must be a congruence from T_1 to T_2 when extended to the initial node. Therefore H is unambiguous. This completes the proof.

For each context limited grammar G let the numbers $l_1(G)$ and $l_2(G)$ be defined as follows.

First, let the <u>length</u> $\lambda(\alpha \to \beta)$ of a production $\alpha \to \beta$ of G be $|\alpha| - 1$, where $|\alpha|$ is the length of α.

Call the production $\alpha \to \beta$ of G a <u>splitting</u> production of G if β contains an external letter. Otherwise call $\alpha \to \beta$ a <u>nonsplitting</u> production. Let

$l_1(G) = \Sigma\{\lambda(P) : \text{P a nonsplitting production of G}\}$

$l_2(G) = \Sigma\{\lambda(P) : \text{P a splitting production of G}\}$

Clearly G is context free just in case $l_1(G) = l_2(G) = 0$.

Call a production P <u>context bound</u> if $\lambda(P) > 0$.

The program is to reduce $l_2(G)$ if it is not 0, and, if $l_2(G) = 0 < l_1(G)$, then to reduce $l_1(G)$. The latter reduction will be considered next.

PROPOSITION 3.4. Let G be a context limited grammar with $l_1(G) > 0$ and $l_2(G) = 0$. Let R be a regular set. Then there exists a context limited grammar H satisfying the following:

(i) $l_1(H) < l_1(G)$.

(ii) $L(H) = L(G)$.

(iii) If G is unambiguous modulo R then H is unambiguous modulo R.

Proof. By Proposition 3.2, we can assume that the partial ordering on the alphabet of G is a linear ordering. Then every set of letters has a highest member, where X higher than Y means Y lower than X.

We first want to identify the set of context free letters of G. These are the letters such that, once one of them appears in a derivation, it must give rise to a sequence of context free productions culminating in an external letter. Thus a context free letter constitutes a barrier that splits the derivation into two independent parts. The set of context free letters is the union of the sets C_i which are defined inductively as follows:

C_o is the set of external letters.

Given C_i, X is in C_{i+1} if both of the following hold:

(i) X is not in α for any context bound production $\alpha \rightarrow \beta$.

(ii) For each production $X \rightarrow \beta_1 Y \beta_2$, where Y is the highest letter in $\beta_1 Y \beta_2$, Y is in C_i.

Note that $C_i \subset C_{i+1}$ for each i.

We define the height of a context free letter X to be the least i such that X is in C_i.

For each context free letter X we define the term <u>tree</u> <u>of</u> X inductively as follows. If X is an external letter, then a tree of X is a labeled production graph with a single node labeled X. Once we have defined the trees of X for all X of height at most n, we extend the definition to letters of height n + 1 as follows. Let X be a context free letter of height n + 1 and let $X \rightarrow \alpha_1 Y \alpha_2$ be a production of G where Y is the highest letter of $\alpha_1 Y \alpha_2$, and Y is not in α_1. Then Y is context free of height at most n. Let p be a node and let $\sigma_1 q \sigma_2$ be a string of nodes such that $\varphi(p) = X$, $\varphi(\sigma_1) = \alpha_1$, $\varphi(q) = Y$, $\varphi(\sigma_2) = \alpha_2$. Let T be a tree of Y with initial string q and no other nodes in common with $(p, \sigma_1 q \sigma_2)$. (Trees of X are production graphs, of course.) Then $(\{(p, \sigma_1 q \sigma_2)\}, q) + T$ is a tree of X.

It is clear that if p is a node of a complete G-graph T with $\varphi(p) = X$, where X is a context free letter, then p is the initial node of a subgraph of T, which is a tree of X. For all of the necessary productions are there, and p is a string of T.

Let us call a letter <u>context</u> <u>bound</u> if it is not context free.

There exists a context bound nonsplitting production $\alpha \rightarrow \beta$ such that some letter in β is context free. For, suppose there were no such production. We will show that, given any context bound letter X, there exists another context bound letter Y that is higher than X. Either X belongs to a context bound production $\alpha_1 X \alpha_2 \rightarrow \beta$, in which case the highest letter Y of β is higher than X and by hypothesis context bound, or else there is a production $X \rightarrow \beta$ in which the highest letter Y of β, again higher than X, is context bound. Hence G could not contain any context bound letter lest there be an infinite ascending

sequence of letters, impossible in a partial ordering on a finite set. But G does contain a context bound letter because $l_1(G)$ is positive, and therefore there is at least one context bound nonsplitting production $\alpha \to \beta$, and our hypothesis has it that the highest letter in β is context bound. Hence the hypothesis is false.

That is, G has a nonsplitting production

$$\alpha \to \beta_1 X \beta_2,$$

where $|\alpha| > 1$, X is not in β_1, and X is context free.

Let $\gamma_1, \ldots, \gamma_k$ be the set of all terminal words of trees of X. Note that all the γ_i are distinct even though two incongruent trees of X may have the same terminal word.

We define the grammar H to be just like G except that H lacks the production $\alpha \to \beta_1 X \beta_2$ and has instead all the productions

$$\alpha \to \beta_1 \gamma_i \beta_2, \; 1 \leq i \leq k.$$

Each γ_i contains an external letter, so each production $\alpha \to \beta_1 \gamma_i \beta_2$ is a splitting production, hence contributes nothing to $l_1(H)$. $\alpha \to \beta_1 X \beta_2$ contributes at least 1 to $l_1(G)$ and is absent from H. Therefore $l_1(H) < l_1(G)$.

Let $T = (\Phi, \mu)$ be a G-graph. Let $T_1 = (\Phi_1, \sigma)$ be a subgraph of T. Call T_1 a <u>removable</u> subgraph of T if $T_1 = (\sigma, \tau_1 p \tau_2) + T_1'$ where T_1' is a tree of X with initial node p, $\varphi(\sigma) = \alpha$, $\varphi(\tau_1) = \beta_1$, $\varphi(p) = X$, $\varphi(\tau_2) = \beta_2$. We are interested in forming an H-graph by replacing all removable subgraphs by productions, so, recalling 2.1.21, we are interested in the separability of two distinct removable subgraphs.

To establish this separability, we need to show that for every context free letter Y, two distinct trees of Y which are subgraphs of T have no productions in common. We proceed by induction on the height of Y. If Y has height 0, then Y is an external letter, hence no tree of Y has any productions at all, so the assertion holds in this case. Suppose the assertion holds for heights at most n, and let Y have height n + 1. Let T_1 and T_2 be trees of Y and subgraphs of T. Let $T_1 = (p_1, \sigma_1' q_1 \sigma_1'') + U_1$ where U_1 has initial node q_1 and is a tree of $\varphi(q_1)$. Let $T_2 = (p_2, \sigma_2' q_2 \sigma_2'') + U_2$ where U_2 has initial node q_2 and is a tree of $\varphi(q_2)$. If the initial productions of T_1 and T_2 are the same then $q_1 = q_2$, because these nodes correspond to the leftmost occurrence of the highest letter of the word. If $\varphi(q_1)$ has height 0, then U_1 and U_2 have no productions, in which case $T_1 = T_2$. If $\varphi(q_1)$ has positive height, then U_1 and U_2 have initial productions with q_1 in the first string of each. Since T can have only one production with q_1 in its first string (2.1.2), by inductive hypothesis U_1 and U_2 are identical. Hence T_1 and T_2 are identical.

Suppose the initial production of T_1 is in U_2. To remove this possibility, let us show that if V is a tree of a context free letter Z and (p, σ) is a noninitial production of V, then $\varphi(p)$ has height less than the height of Z. The case of height 0 is trivial, and if the assertion holds for heights up to n, and Z has height n + 1, then $V = (q, \tau) + V'$ where V' is a tree of a letter Z' of smaller height than Z. (p, σ) is either the initial production of V', in which case $\varphi(p) = Z'$ has smaller height than Z, or else (p, σ) is a noninitial production of V', in which case $\varphi(p)$ has smaller height than Z', hence smaller height than Z. Thus if the initial production of T_1 is in U_2, we are forced to

the conclusion that Y has smaller height than Y. For the same reason, the initial production of T_2 is not in U_1.

With these facts in mind, suppose (p,σ) is a production belonging to both T_1 and T_2, with $\varphi(p)$ having the largest possible height consistent with this property. (p,σ) is not the initial production of either T_1 or T_2. Hence T_1 contains a production (p',σ') with p in σ' and $\varphi(p')$ of greater height than $\varphi(p)$. But p cannot appear in the second strings of two distinct productions, so T_2 must also have the production (p',σ'). This contradicts the maximal choice of $\varphi(p)$.

We conclude that two trees of Y, subgraphs of T, are either identical or share no productions.

Now we can show that two distinct removable subgraphs have no productions in common. Let T_1 and T_2 be removable subgraphs of T. We can say

$$T_1 = (\sigma_1, \tau_1' p_1 \tau_1'') + U_1 \text{ and } T_2 = (\sigma_2, \tau_2' p_2 \tau_2'') + U_2$$

where $\varphi(\sigma_1) = \varphi(\sigma_2) = \alpha$, $\varphi(\tau_1') = \varphi(\tau_2') = \beta_1$, $\varphi(p_1) = \varphi(p_2) = X$, $\varphi(\tau_1'') = \varphi(\tau_2'') = \beta_2$, and U_1 and U_2 are trees of X with initial nodes p_1 and p_2 respectively. If $p_1 = p_2$, then U_1 and U_2 must have the same initial production, since there is only one production with p_1 in its first string. Hence U_1 and U_2 are identical by the assertion proved previously. Also, if $p_1 = p_2$, since p_1 can be in the second string of only one production,

$$(\sigma_1, \tau_1' p_1 \tau_1'') = (\sigma_2, \tau_2' p_2 \tau_2'').$$

Therefore to have T_1 and T_2 distinct, we need $p_1 \neq p_2$.

$(\sigma_1, \tau_1' p_1 \tau_1'') \neq (\sigma_2, \tau_2' p_2 \tau_2'')$ because p_1 and p_2 are required to correspond to the leftmost occurrence of X in $\varphi(\tau_1' p_1 \tau_1'')$ and $\varphi(\tau_2' p_2 \tau_2'')$ respectively, so that

the equality of the two strings would entail $p_1 = p_2$. Neither initial production of one is a noninitial production of the other, for both initial productions are context bound and all noninitial productions are context free. U_1 and U_2 are distinct trees of X and therefore have no productions in common. Therefore T_1 and T_2 have no productions in common.

T_1 has initial string σ_1 and just one initial production $(\sigma_1, \tau_1' p_1 \tau_2')$, hence no initial node of T_1 is terminal. Similarly, no initial node of T_2 is terminal, and T_2 has only one initial production. Therefore, by 2.1.22, T_1 and T_2 are separable.

Now in a complete G-graph, every production (σ, τ) with label $\alpha \rightarrow \beta_1 X \beta_2$ must belong to a removable subgraph, because the node labeled X must be the initial node of a tree of X, as noted earlier, and σ is a string of T by 2.1.18. Therefore if we replace all the removable subgraphs with productions of H, we will have an H-graph.

More precisely, let $T_1, \ldots, T_k$ be all the removable subgraphs of T, with initial strings $\sigma_1, \ldots, \sigma_k$ and terminal strings $\tau_1, \ldots, \tau_k$. For each j the terminal word of T_j is $\varphi(\tau_j) = \beta_1 \gamma_{i_j} \beta_2$. Let the production graph U_j be $(\{(\sigma_j, \tau_j)\}, \sigma_j)$. Each U_j is an H-graph, for the only production label in U_j is $\alpha \rightarrow \beta_1 \gamma_{i_j} \beta_2$. No U_j has any noninitial nonterminal nodes, so the U_j contain no noninitial nonterminal nodes in common with each other or with T. Therefore $T_1, \ldots, T_k$ are replaceable by $U_1, \ldots, U_k$ in T.

Denote by F(T) the graph formed by replacing $T_1, \ldots, T_k$ with $U_1, \ldots, U_k$ in T.

By 2.1.16 and 2.1.21, F(T) has the same initial and terminal strings as T, and by definition its labeling agrees with that of T on the initial and terminal

nodes. Hence F(T) has the same initial and terminal words as T. F(T) is an H-graph.

We note in passing that this shows that $L(G) \subset L(H)$.

Suppose h is a congruence defined on T. Let us show that F(h(T)) is congruent to F(T). $h(T_1),\ldots,h(T_k)$ are all the removable subgraphs of h(T , their initial strings are $h(\sigma_1),\ldots,h(\sigma_k)$ respectively, and their terminal strings are $h(\tau_1),\ldots,h(\tau_k)$ respectively. Clearly $(\{(\sigma_j,\tau_j)\},\sigma_j)$ is congruent to $(\{(h(\sigma_j),h(\tau_j))\},h(\sigma_j))$. That is, U_j is congruent to $h(U_j)$. F(h(T)) is the graph formed by replacing $h(T_1),\ldots,h(T_k)$ by $h(U_1),\ldots,h(U_j)$ in h(T). We have the situation described in 2.2.3. Therefore F(T) is congruent to F(h(T)).

Conversely, let T be a complete H-graph. Let $(\sigma_1,\tau_1),\ldots,(\sigma_k,\tau_k)$ be all of the productions of T having labels of the form $\alpha \rightarrow \beta_1\gamma_1\beta_2$. That is, $\varphi(\sigma_j) = \alpha$, $\varphi(\tau_j) = \beta_1\gamma_{1_j}\beta_2$. For each j construct the graph

$$T_j = (\{(\sigma_j,\rho'_j q_j \rho''_j)\},\sigma_j) + V_j,$$

where $\varphi(\rho'_j) = \beta_1$, $\varphi(q_j) = X$, $\varphi(\rho''_j) = \beta_2$, V_j is a tree of X with initial string q_j, and terminal word γ_{1_j}. Moreover, we choose ρ'_j, ρ''_j and the terminal string of V_j so that the terminal string of T_j is τ_j. We choose the rest of the nodes of the V_j to be distinct from all the nodes of T and so that no two V_j have nodes in common other than initial or terminal nodes.

Any two productions (σ_i,τ_i) and (σ_j,τ_j), considered as subgraphs, are separable.

Therefore $(\sigma_1,\tau_1),\ldots,(\sigma_k,\tau_k)$, considered as graphs, are replaceable by $T_1,\ldots,T_k$ in T. Let T' be the graph formed by the replacement of the former by the latter in T. T' is a G-graph because all production labels of the form $\alpha \rightarrow \beta_1\gamma_1\beta_2$ have been removed. T' has the same initial and terminal words as T.

(P1) $X \to \beta_1 b$,

(P2) $\alpha \to c\beta_2$.

The nonsplitting productions of H are exactly the nonsplitting productions of G, so $l_1(H) = l_1(G)$. The splitting productions of H are just those of G except that H has $\alpha \to c\beta_2$ and $X \to \beta_1 b$ instead of $X\alpha \to \beta_1 a\beta_2$. The length of $X \to \beta_1 b$ is 0, and the length of $\alpha \to c\beta_2$ is $|\alpha| - 1$, whereas the length of $X\alpha \to \beta_1 a \beta_2$ is $|\alpha|$. Hence $l_2(H) < l_2(G)$.

Let V be the alphabet of G. Let

$$R' = V^* \cup (V^* \, bc \, V^*)^*.$$

That is, R' is the set of all words on $V \cup \{b,c\}$ such that b and c occur only in subwords bc. Clearly R' is regular.

Let M be a sequential machine which replaces any $Y \neq b,c$ by Y, and replaces bc by a. Let

$$R_2 = M^{-1}(R_1) \cap R'.$$

$M^{-1}(R_1)$ is regular, as is proved in [5, p.186], and it is well known that the intersection of two regular sets is regular. Therefore R_2 is regular.

We will use, merely as a notational device, the grammar H', which is just like H except that H' has the production $bc \to a$. We also define a graph U corresponding to the productions $X \to \beta_1 b$, $\alpha \to c\beta_2$, $bc \to a$. More precisely,

$$U = (\{p_U, \tau_U q_U), (\sigma_U, r_U \rho_U), (q_U r_U, s_U)\}, p_U \sigma_U)$$

where $\varphi(p_U) = X$, $\varphi(\sigma_U) = \alpha$, $\varphi(\tau_U) = \beta_1$, $\varphi(q_U) = b$, $\varphi(\rho_U) = \beta_2$, $\varphi(r_U) = c$, $\varphi(s_U) = a$. U is an H'-graph.

I claim that if an H'-graph T such that b and c do not occur in the initial word of T has a production (qr, s) with $\varphi(q) = b$, $\varphi(r) = c$, $\varphi(s) = a$, then the

Let us note in passing that this shows $L(H) \subset L(G)$. Therefore, since we have the other inclusion, $L(H) = L(G)$.

$T_1, \ldots, T_k$ are all of the removable subgraphs of T, for the production label $\alpha \rightarrow \beta_1 X \beta_2$ can be in T' only by being in one of the T_j. The T_j have initial strings σ_j and terminal strings τ_j. Therefore it is clear that $F(T') = T$.

Now let T_1 and T_2 be two complete H-graphs with the same terminal word w where w is in R. There exist two complete G-graphs, T_1' and T_2', with terminal word w, such that $F(T_1') = T_1$ and $F(T_2') = T_2$. If G is unambiguous modulo R, then T_1' and T_2' must be congruent. We have seen that this implies that $F(T_1')$ is congruent to $F(T_2')$.

This establishes (iii) and finishes the proof.

PROPOSITION 3.5. Let G be a context limited grammar. Let $l_2(G) > 0$. Let R_1 be a regular set. There exists a context limited grammar H, a sequential machine [3, p.93] M, and a regular set R_2 satisfying the following:

(i) $L(G) \cap R_1 = M(L(H) \cap R_2)$.

(ii) If G is unambiguous modulo R_1, then H is unambiguous modulo R_2.

(iii) If G is unambiguous modulo R_1, then M is one-to-one on $L(H) \cap R_2$.

(iv) $l_1(H) = l_1(G)$ and $l_2(H) < l_2(G)$.

Proof. Since $l_2(G) > 0$, G has a splitting production $X\alpha \rightarrow \beta_1 a \beta_2$ where a is an external letter and $|\alpha| > 0$.

Let b and c be letters not in the alphabet of G. Let H be exactly like G except that H has b and c in its external alphabet, and H lacks the production $X\alpha \rightarrow \beta_1 a \beta_2$ and has instead the productions

production (qr,s) belongs to one and only one subgraph of T which is congruent to U. For, q cannot be an initial node of T, so q must be in the second string of some production of T. Such a production must have the form $(p,\tau q)$ with $\varphi(p) = X$, $\varphi(\tau) = \beta_1$. Similarly, there must be a production $(\sigma, r\rho)$ with $\varphi(\sigma) = \alpha$, $\varphi(\rho) = \beta_2$. Now

$$T_1 = (\{(p,\tau q), (\sigma, r\rho), (qr,s)\}, p\sigma)$$

is a labeled graph congruent to U. We could not find another such congruent image of U lest ρ or q appear in the second strings of two distinct productions of T. T_1 is a subgraph of T provided its terminal string $\tau s\rho$ is a string of T (2.1.15). τq, $r\rho$, and qr are strings of T because they are strings of productions of T, hence by two applications of 2.1.9 we learn that $\tau qr\rho$ is a string of T. (qr,s) being a production, $\tau s\rho$ must also be a string of T, hence T_1 is a subgraph of T.

Also, if T_2 is a congruent image of U and a subgraph of T, and if T_2 is distinct from T_1, then T_1 and T_2 are separable. To see this, let V be the smallest initial subgraph of T containing all the productions of T_1. (For the existence of V, see 2.1.17.) Then the terminal productions of V are just the terminal productions of T_1. Now all the productions of T_1 are terminal in T_1. Therefore the terminal productions of V are just the productions of T_1. Removing just these productions yields the initial subgraph of T, which is called $V - T_1$. Now T_1 and T_2 have no productions in common. For, if they did, then either q or r would have to be a node of T_2, since every production of T_1 contains either q or r. So T_2, being congruent to U, would have to contain a production with q or r in the first string. There is only one such production

in T, namely (qr,s). We have already seen that T_1 is the only subgraph of T congruent to U which contains (qr,s), and therefore T_1 and T_2 have no productions in common. Therefore, either V - T_1 contains all the productions of T_2, or none of them. In the former case, V - T_1 establishes the asserted separability; in the latter case, V establishes it.

Let T' be an H'-graph, where b and c occur neither in the initial nor in the terminal word of T'. Let $T_1,\ldots,T_k$ be all the subgraphs of T' which are congruent to U. If (σ,τ) is a production of T' and $\varphi(\sigma) \rightarrow \varphi(\tau)$ is not in G, then (σ,τ) must belong to one of the T_i. For, if $\varphi(\sigma) \rightarrow \varphi(\tau)$ is not in G, then $\varphi(\sigma)$ or $\varphi(\tau)$ contains b or c. If, for example, $\tau = \tau'q$ with $\varphi(q) = B$, then, since b is not in the terminal word of T', there must be a production (qr,s) with $\varphi(r) = c$, $\varphi(s) = a$. (qr,s) belongs to exactly one of the T_i, and that T_1 contains a production with q occurring in the second string. Since there is only one production with q in its second string, $(\sigma,\tau'q) = (\sigma,\tau)$ must be in T_i. The other cases are similar.

It follows that if we replace $T_1,\ldots,T_k$ by G-graphs $U_1,\ldots,U_k$ in T', then the resulting graph will be a G-graph. Letting σ_i and τ_i be respectively the initial and terminal strings of T_i for each i, then $U_i = (\{(\sigma_i,\tau_i)\},\sigma_i)$ is a G-graph for each i, for the only production label of U_i is $\alpha \rightarrow \beta_1 a\beta_2$. The graph formed by replacing $T_1,\ldots,T_k$ by $U_1,\ldots,U_k$ in T' (2.1.21) is a G-graph with the same initial and terminal words as T'. Call this G-graph $F'(T')$.

Conversely, given a G-graph T we can construct an H'-graph T' such that $T = F'(T')$. For, let $(\sigma_1,\tau_1),\ldots,(\sigma_k,\tau_k)$ be all the productions of T such that $\varphi(\sigma_i) = X\alpha$, $\varphi(\tau_i) = \beta_1 a\beta_2$. These productions, considered as graphs, constitute

separable subgraphs of T, that is (σ_i, τ_i) and (σ_j, τ_j) considered as graphs are separable if $i \neq j$. With the usual precautions about distinctness of new nodes, we construct graphs $T_1, \ldots, T_k$ all congruent to U, where each T_i has initial string σ_i and terminal string τ_i. The graph formed by replacing $(\sigma_1, \tau_1), \ldots, (\sigma_k, \tau_k)$ by $T_1, \ldots, T_k$ in T is an H'-graph T'. The only subgraphs of T' congruent to U are $T_1, \ldots, T_k$. T' has, by 2.1.21, the same initial and terminal words as T, hence T' contains neither b nor c in its initial or terminal word, because b and c are not in the alphabet of G. It is clear by inspection of the construction of F' that $F'(T') = T$. By 2.2.3, it is clear from this construction that if $F'(T'')$ is also T, then T'' is congruent to T'.

From this it follows that $L(H') = L(G)$.

Suppose that G is unambiguous modulo the regular set R_1. Then H' is unambiguous modulo R_1. For, suppose T_1 and T_2 are two H'-graphs with the same terminal word w where w is in R_1, and where b and c do not occur in the initial or terminal word of either. If $F'(T_1)$ is congruent to $F'(T_2)$, then T_1 is congruent to T_2. If T_1 and T_2 are complete H'-graphs, then $F'(T_1)$ and $F'(T_2)$ are complete G-graphs and hence must be congruent. Therefore H' is unambiguous modulo R_1.

Now let us show that $L(H') = M(L(H) \cap R')$. Let T be an H-graph with terminal word in R'. There exist k disjoint terminal subwords $p_1q_1, \ldots, p_kq_k$ with $\varphi(p_i) = b$, $\varphi(q_i) = c$ for each i, and such that any node r with $\varphi(r) = b, c$ is one of the p_i or q_i. T is an H'-graph, and if we adjoin new nodes $r_1, \ldots, r_k$, new productions (p_iq_i, r_i), such that $\varphi(r_i) = a$, the resulting graph

T' is an H'-graph. Define $T' = F(T)$. If the terminal word of T is w, then the terminal word of T' is $M(w)$. Thus $M(L(H) \cap R') \subset L(H')$. Conversely, any H'-graph without b,c in the terminal word contains $k \geq 0$ terminal productions $(p_1q_1,r_1),\ldots,(p_kq_k,r_k)$ with $\varphi(p_i) = b$, $\varphi(q_i) = c$, $\varphi(r_i) = a$. Deleting these productions gives an H-graph with terminal word w in R', and $M(w)$ is the terminal word of T'. So $L(H') = M(L(H) \cap R')$.

Now we can verify (i). $L(G) \cap R_1 = M(L(H) \cap R') \cap R_1$. If a word w is in $M(L(H) \cap R_2)$, then $w = M(v)$ for some v in $L(H) \cap R_2$. $L(H) \cap R_2 = L(H) \cap M^{-1}(R_1) \cap R'$, so $M(v) = w$ is in R_1. v is also in $L(H) \cap R'$ so $M(v)$ is in $M(L(H) \cap R')$. Thus w is in $M(L(H) \cap R') \cap R_1$. Conversely, if w is in $M(L(H) \cap R') \cap R_1$, then $w = M(v)$ for some v in $L(H) \cap R'$. Since $M(v)$ is in R_1, v is also in $M^{-1}(R_1)$. So v is in $L(H) \cap R' \cap R_1 = L(H) \cap R_2$. So $M(v) = w$ is in $M(L(H) \cap R_2)$. Thus $M(L(H) \cap R') \cap R_1 = M(L(H) \cap R_2)$, and this with the previous relation implies $L(G) \cap R_1 = M(L(H) \cap R_2)$.

To verify (ii), we already know that if G is unambiguous modulo R_1, then so is H'. We want to conclude that H is unambiguous modulo $M^{-1}(R_1) \cap R'$. Let T_1 and T_2 be complete H-graphs with the same terminal word w where w is in R' and $M(w)$ is in R_1. $F(T_1)$ and $F(T_2)$ are complete H'-graphs, each with terminal word $M(w)$. Since H' is unambiguous modulo R_1, $F(T_1)$ is congruent to $F(T_2)$. But T_1 and T_2 are formed from $F(T_1)$ and $F(T_2)$ respectively by deleting productions that correspond in the congruence. Therefore T_1 is congruent to T_2. Therefore H is unambiguous modulo R_2.

To verify (iii), suppose G is unambiguous modulo R_1, and let w_1 and w_2 be two words in $L(H) \cap R_2$. We want to prove that $M(w_1) \neq M(w_2)$ if $w_1 \neq w_2$.

Let T_1 and T_2 be complete H-graphs with terminal words w_1 and w_2 respectively. Then $F(T_1)$ and $F(T_2)$ are complete H'-graphs with terminal words $M(w_1)$ and $M(w_2)$ respectively. Since w_1 and w_2 are in $R_2 = M^{-1}(R_1) \cap R'$, $M(w_1)$ and $M(w_2)$ are in R_1. If $M(w_1) = M(w_2)$, then by the unambiguity modulo R_1 of H', $F(T_1)$ is congruent to $F(T_2)$. We noted before that this entails T_1 being congruent to T_2. But that would require $w_1 = w_2$. Therefore M is one-to-one on $L(H) \cap R_2$.

This finishes the proof.

PROPOSITION 3.6. Let G be a context limited grammar and R a regular set. Then $L(G) \cap R$ is a context free language. Moreover, if G is unambiguous modulo R, then $L(G) \cap R$ is unambiguous as a context free language.

<u>Proof</u>. Suppose there is a counterexample. Let x be the smallest integer for which there is a counterexample G' such that $l_1(G') = x$. Let y be the smallest integer for which there is a counterexample G with $l_1(G) = x$ and $l_2(G) = y$.

Suppose $y > 0$. Let R_1 be a regular set. G and R_1 satisfy the hypothesis of 3.5. So let M, H and R_2 be as in the conclusion of 3.5. Then $l_2(H) < y$ and $l_1(H) = x$, so $L(H) \cap R_2$ is a context free language. Sequential machine mappings preserve context freedom [5, p.184], so $M(L(H) \cap R_2)$ is context free. Thus $L(G) \cap R_1$ is a context free language. If G is unambiguous modulo R_1, then H is unambiguous modulo R_2. Again, because H satisfies the present proposition, $L(H) \cap R_2$ is an unambiguous context free language. Moreover, M is one-to-one on $L(H) \cap R_2$, so that M produces an unambiguous image of $L(H) \cap R_2$ [6, Theorem 1]. So $L(G) \cap R_1$ is unambiguous. But this would mean that G is not a counterexample. Therefore $y = 0$.

Suppose $x > 0$. Let R be a regular set. G satisfies the hypothesis of 3.4. Let H be as in the conclusion of 3.4. Then $l_1(H) < x$. So $L(H) \cap R$ is a context free language. But $L(H) = L(G)$, so $L(G) \cap R$ is a context free language. Moreover, if G is unambiguous modulo R, then H is unambiguous modulo R, and since H is not a counterexample, $L(H) \cap R$ must be unambiguous, i.e., $L(G) \cap R$ is unambiguous, and we have contradicted the assumption that G is a counterexample.

Therefore x and y are both 0. But this means that G is already a context free grammar, hence $L(G) \cap R$ is a context free language for every regular set R, for intersection with regular sets preserves context freedom [1, p.171]. By 3.3, if G is unambiguous modulo R, then $L(G) \cap R$ is unambiguous. Again we have to deny that G is a counterexample, so there must not be any counterexamples.

Letting $R = \Sigma^*$ where Σ is the external alphabet of G, we have immediately from 3.6, 3.1, and 2.2.4:

THEOREM 1. <u>The context limited languages are</u> (<u>disregarding</u> ϵ) <u>just the context free languages</u>.

THEOREM 2. <u>The unambiguous context limited languages are</u> (<u>disregarding</u> ϵ) <u>just the unambiguous context free languages</u>.

REFERENCES

[1] Bar-Hillel, Y., Perles, M., and Shamir, E. On formal properties of simple phrase structure grammars. Zeit Phonetik, Sprachwiss Kommunik 14, 1961, pp. 143-172.

[2] Evey, R. J. The theory and application of pushdown store machines. Mathematical Linguistics and Automatic Translation, Report No. NSF-10 to the National Science Foundation, The Computation Laboratory of Harvard University, 1963.

[3] Ginsburg, S. The mathematical theory of context free languages. McGraw-Hill, New York, 1966.

[4] Ginsburg, S. and Greibach, S. Mappings which preserve context sensitive languages. SDC document TM-738/018/00, 1965.

[5] Ginsburg, S. and Rose, G. Operations which preserve definability in languages. Journal of the Association for Computing Machinery, Vol. 10, 1965, pp. 175-195.

[6] Ginsburg, S. and Ullian, J. S. Preservation of unambiguity and inherent ambiguity in context free languages. Journal of the Association for Computing Machinery, Vol. 13, 1966, pp. 364-368.

[7] Greibach, S. A. A new normal form theorem for context free phrase structure grammars. Journal of the Association for Computing Machinery, Vol. 13, 1965, pp. 42-52.

[8] Greibach, S. A. A standard form theorem for phrase structure generators. Mathematical Linguistics and Automatic Translation, Report No. NSF-13 to the National Science Foundation, The Computation Laboratory of Harvard University, 1964.

[9] Griffiths, T. V. Turing machine recognizers for general rewriting systems. Proceedings of the Fifth Annual Symposium on Switching Circuit Theory and Logical Design. The Institute of Electrical and Electronics Engineers, 1964, pp. 47-56.

[10] Hibbard, T. N. Scan limited automata and context limited grammars. Doctoral dissertation, University of California, Los Angeles, 1966.

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) System Development Corporation Santa Monica, California	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP
3. REPORT TITLE THE EQUIVALENCE OF CONTEXT LIMITED GRAMMARS TO CONTEXT FREE GRAMMARS	
4. DESCRIPTIVE NOTES (Type of report and inclusive dates) Scientific Report Interim 4	
5. AUTHOR(S) (Last name, first name, initial) Hibbard, Thomas N.	

6. REPORT DATE 16 January 1967	7a. TOTAL NO. OF PAGES 63	7b. NO. OF REFS 10
8a. Contract F1962867C-0008 and Grant No. AF-AFOSR-1203-67, Programming (Algorithmic) Languages. Project, Task, Work Unit Nos. 5632 - 05 - 01 DoD Element 61445014 DoD Subelement 681305	9a. ORIGINATOR'S REPORT NUMBER(S) TM-738/030/00 Scientific Report No. 4 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) AFCRL-67-0434	

10. AVAILABILITY/LIMITATION NOTICES

Distribution of this document is unlimited. It may be released to the Clearinghouse, Department of Commerce, for sale to the general public.

11. SUPPLEMENTARY NOTES Hq. AFOSR, OAR (SRIR) U. S. Air Force Arlington, Virginia	12. SPONSORING MILITARY ACTIVITY Air Force Cambridge Research Laboratories (CRB) L. G. Hanscom Field, Bedford, Mass. 01730

13. ABSTRACT

A grammar G is context limited if there exists a partial ordering on the alphabet of G under which, for every production $\alpha \to \beta$ of G, every letter of α is smaller than some letter of β. It is proved that the languages generated by contect limited grammars are just the context free languages. Unambiguity of general grammars is defined and discussed carefully, preparatory to proving that the languages generated by unambiguous context limited grammars are just the unambiguous context free languages.

DD FORM 1 JAN 64 1473

Unclassified

Security Classification

Unclassified
Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Context Limited Grammars Context Free Languages						

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization *(corporate author)* issuing the report.

2a. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2b. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year; or month, year. If more than one date appears on the report, use date of publication.

7a. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7b. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8a. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8b, 8c, & 8d. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9a. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9b. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers *(either by the originator or by the sponsor)*, also enter this number(s).

10. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through ______."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through ______."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through ______."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring *(paying for)* the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS)*, *(S)*, *(C)*, or *(U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, rules, and weights is optional.

Unclassified
Security Classification